

**PREDIKSI KREDIT MACET BERDASARKAN PREFERENSI
NASABAH MENGGUNAKAN METODE KLASIFIKASI C4.5
PADA KOPERASI SIMPAN PINJAM MITRA RAYA WATES**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Iqbal Taufiq Ahmad Nur

NIM: 145150400111009



PROGRAM STUDI SISTEM INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

PREDIKSI KREDIT MACET BERDASARKAN PREFERENSI NASABAH MENGGUNAKAN
METODE KLASIFIKASI C4.5 PADA KOPERASI SIMPAN PINJAM MITRA RAYA WATES

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Iqbal Taufiq Ahmad Nur
NIM: 145150400111009


Skripsi ini telah diuji dan dinyatakan lulus pada
26 Juli 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

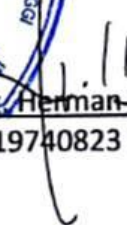

Nanang Yudi Setiawan, S.T., M.Kom.
NIP. 19760619 200604 1 001


Dr. Eng. Fitra A. Bachtiar, S.T., M.Eng
NIK. 201201 840628 1 001

Mengetahui

Ketua Jurusan Sistem Informasi




Dr. Eng. Herman Tolle, S.T., M.T.
NIP. 19740823 200012 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 26 Juli 2018



Iqbal Taufiq Ahmad Nur

NIM: 145150400111009

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, karena berkat Rahmat dan Hidayah-Nya, penulis dapat menyelesaikan laporan skripsi yang berjudul “Prediksi Kredit Macet Berdasarkan Preferensi Nasabah Menggunakan Metode Klasifikasi C4.5 pada Koperasi Simpan Pinjam Mitra Raya Wates”.

Penulis mengucapkan terimakasih kepada pihak-pihak yang telah membantu penulis dalam menyelesaikan skripsi ini, antara lain:

1. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya.
2. Bapak Dr. Eng. Herman Tolle, S.T., M.T. selaku Ketua Jurusan Sistem Informasi Fakultas Ilmu Komputer Universitas Brawijaya.
3. Bapak Suprpto, S.T., M.T. selaku Ketua Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Brawijaya.
4. Bapak Nanang Yudi Setiawan, S.T., M.Kom. selaku Dosen Pembimbing I yang telah memberikan waktu, bimbingan, ilmu, arahan, nasihat, dan masukan untuk penyelesaian skripsi ini.
5. Bapak Dr. Eng. Fitra A. Bachtiar, S.T, M.Eng selaku Dosen Pembimbing II yang telah memberikan waktu, bimbingan, ilmu, arahan, nasihat, dan masukan untuk penyelesaian skripsi ini.
6. Bapak Himawat Aryadita, S.T., M.Sc. selaku Dosen Pembimbing Akademik penulis yang telah memberikan bimbingan selama kuliah.
7. Bapak Alex Franico selaku Pimpinan dan juga seluruh karyawan KSP Mitra Raya Wates yang tidak dapat disebutkan satu persatu oleh penulis atas bantuan yang telah diberikan selama penulis melakukan penelitian.
8. Kedua Orang Tua, Bapak M. Abbas, Ibu Tatik Nurur R., kedua adik Kiandra Aulia P. dan Risang Rizky Akbar A. yang selalu memberikan motivasi, kasih sayang serta dukungan moril serta materiil.
9. Om Edy Santoso, Tante Rullin Rodiah, adik sepupu Tiara Rulita A., Bayu Priyatama, dan Anastasya Fitri A. yang telah berperan sebagai keluarga kedua di Malang serta selalu memberikan motivasi, kasih sayang serta dukungan moril serta materiil.
10. Seluruh keluarga penulis, yang selalu memberikan doa dan dukungan dalam pengerjaan skripsi.
11. Ferlie Deanada E. selaku teman dekat yang selalu menjadi tempat bercerita dan berkeluh kesah, menemani, menyemangati, mendoakan, dan memberikan nasihat kepada penulis dan membantu penulis ketika penulis sedang dalam kesulitan dalam beberapa hal.

12. Vicky Nur Ardianto, Dini Indah N.R.P.R., Oggy Setiawan, Anandya Femia W., Jawara Wahyu A., Mashuda Bahtiar, Savira Fahrurisa, Iqbal Caraka A., Andry Lesmana J., Ali Fikri, M. Yogi Saka F., Tedja Adhi dan Helmi Danu P. selaku sahabat dan teman yang selalu memberikan bantuan dalam hal dukungan moril dan materil serta menjadi tempat berkeluh kesah dan tempat bertukar pikiran yang secara tidak langsung telah membantu penulis menyelesaikan skripsi ini.
13. Teman-teman “Gerakan 10 September”, “Tim Bujang”, “Tim Manut”, “Divisi Acara Kepanitiaan Dies Natalies PTIIK ke-4”, “Divisi Acara Kepanitiaan Olimpiade Filkom 2016”, “Divisi Acara i-Fest 3.0” yang tidak bisa penulis sebutkan satu persatu atas pengalaman dan waktu yang sangat berharga selama penulis menempuh pendidikan hampir empat tahun di Fakultas Ilmu Komputer.
14. Teman-teman lainnya yang tidak dapat disebutkan satu persatu, yang telah memberikan dukungan, doa, dan semangat untuk penulis.
15. Seluruh pihak yang telah membantu secara langsung maupun tidak langsung kepada penulis sehingga penulis dapat menyelesaikan skripsi ini.

Penulis menyadari bahwa dalam penyusunan laporan skripsi ini masih banyak kekurangan baik format laporan maupun isinya. Untuk itu penulis sangat mengharapkan kritik dan saran yang membangun demi perbaikan selanjutnya. Semoga laporan skripsi ini dapat bermanfaat bagi pembaca dan penulis, aamiin.

Malang, 26 Juli 2018

Penulis

iqbaltaufiq.an@gmail.com

ABSTRAK

Kredit macet merupakan permasalahan utama yang sedang dihadapi lembaga finansial khususnya koperasi di Indonesia. Hal tersebut juga dialami pada KSP Mitra Raya Wates yang tidak menggunakan tenaga analis kredit dan pengambilan keputusannya menggunakan pendekatan intuitif dan berdasarkan pengalaman yang sudah ada oleh Pimpinan KSP. Hal tersebut tentu akan menimbulkan adanya kesalahan analisis kredit yang dapat menyebabkan risiko kredit. Proses survei yang dilakukan pada KSP Mitra Raya Wates juga tidak bisa menjamin pinjaman yang dilakukan oleh nasabah bebas dari risiko kredit, mengingat terdapat kurang lebih 100 orang nasabah mengalami kredit macet dari total kurang lebih 400 nasabah yang telah mendapatkan pinjaman. Sehingga diperlukan sistem yang mampu menjadi pendukung keputusan guna mendeteksi kualitas kredit sejak dini. Klasifikasi merupakan salah satu metode yang dapat digunakan untuk memprediksi kualitas kredit nasabah. Dengan memanfaatkan salah satu metode klasifikasi yaitu C4.5 akan dihasilkan sebuah *rule* dalam bentuk pohon keputusan yang dapat digunakan untuk memprediksi kredit macet dalam kasus pengajuan pinjaman nasabah. Data yang terkumpul berjumlah 193 data yang kemudian dilakukan *pre processing* dan pembagian data untuk kemudian dilakukan proses pembentukan *rule* dalam bentuk pohon keputusan menggunakan *tool weka*. Sistem kemudian diimplementasikan dengan memanfaatkan *framework codeigniter* dan *simple cli weka*. Hasil yang didapatkan dari pengevaluasian dan pemvalidasian menggunakan *confussion matrix* adalah tingkat akurasi sebesar 94,5946%. Sedangkan berdasarkan kurva ROC dihasilkan nilai AUC sebesar 0,9689. Tingkat *usability system* yang dihasilkan dengan memanfaatkan SUS adalah sebesar 82,5. Luaran yang dihasilkan berupa visualisasi *dashboard* dengan beberapa grafik yang memuat persentase, *time-series* dan *trend* dari total pengajuan yang telah dilakukan dan juga *form* yang dapat digunakan oleh pihak KSP Mitra Raya Wates untuk melakukan prediksi pengajuan kredit nasabah dan juga pemasukan *dataset* ke sistem.

Kata kunci: *kredit, prediksi, klasifikasi, C4.5, weka, codeigniter, SUS*

ABSTRACT

Bad credit is the main problem that faced by financial institutions, especially cooperatives in Indonesia. This problem is also happened in KSP Mitra Raya Wates that does not use credit analyst and the decision making process is using an intuitive approach and based on existing experience that owned by KSP Leader. The problem will certainly lead to a mistake in credit analysis that may cause credit risk. The survey process conducted at KSP Mitra Raya Wates also cannot guarantee that the loans made by customers are free from credit risk, considering there are approximately 100 customers having bad credit from a total of approximately 400 customers who have received loans. KSP Mitra Raya Wates needs a system that capable of supporting decision to detect credit quality early on. Classification is one of the methods that can be used to predict credit quality of customers. By utilizing C4.5 classification method, it will generate a rule in the form of a decision tree that can be used to predict bad credit in the case of customer loan application. 193 data had been collected and then will apply preprocessing and data sharing, after that the process of forming the rule in the form of decision tree using weka tool will be done. Then the system will be implemented by utilizing the codeigniter framework and simple cli weka. The results of evaluating and validating using confusion matrix have accuracy of 94.5946%. While based on the ROC curve, it generated AUC value of 0.9689. The level of system usability generated by utilizing SUS is 82.5%. The output is dashboard visualization with several graphs containing the percentage, time-series and trend of total submissions that have been made and also forms that can be used by KSP Mitra Raya Wates to make predictions of customer credit application and also dataset entry into the system.

Keywords: credit, prediction, clasification, C4.5, weka, codeigniter, SUS

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xv
DAFTAR KODE.....	xviii
DAFTAR LAMPIRAN	xix
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	3
1.3 Tujuan	3
1.4 Manfaat.....	4
1.5 Batasan masalah	4
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Kajian Pustaka	6
2.2 Profil KSP Mitra Raya	11
2.3 Kredit.....	12
2.4 <i>Knowledge Discovery in Database (KDD) dan Data Mining</i>	12
2.4.1 <i>Knowledge Discovery in Database (KDD)</i>	12
2.4.2 <i>Data Mining</i>	13
2.5 Klasifikasi.....	14
2.6 <i>Decision Tree</i>	15
2.6.1 Algoritme C4.5.....	15
2.6.2 <i>Pruning dalam Decision Tree</i>	16
2.7 Metode Evaluasi dan Validasi Algoritme <i>Data Mining</i>	17
2.7.1 <i>Confussion Matrix</i>	17

2.7.2 Kurva ROC.....	18
2.8 Weka	18
2.9 Analisis Kebutuhan Sistem.....	18
2.10 Perancangan Sistem.....	19
2.11 UML (<i>Unified Modelling Language</i>).....	20
2.11.1 Use Case Diagram	20
2.11.2 Activity Diagram.....	21
2.11.3 Sequence Diagram	21
2.11.4 Class Diagram.....	22
2.12 Implementasi Sistem	23
2.12.1 Object Oriented Programming (OOP)	23
2.12.2 Hyper Text Markup Language (HTML)	24
2.12.3 Cascading Style Sheet (CSS).....	24
2.12.4 Hypertext Preprocessor (PHP)	24
2.12.5 Codeigniter	25
2.13 Black-Box Testing	25
2.13.1 Validation Testing	26
2.13.2 Usability Testing.....	26
2.14 Konsep Dasar <i>Dashboard</i>	28
2.14.1 Definisi <i>Dashboard</i>	28
2.14.2 Tujuan Penggunaan <i>Dashboard</i>	28
2.15 Snake Case	29
BAB 3 METODOLOGI	30
3.1 Identifikasi Masalah	30
3.2 Studi Literatur	30
3.3 Analisis Kebutuhan	31
3.4 Pengumpulan Data	31
3.5 Perancangan	31
3.6 Implementasi	33
3.7 Pengujian dan Analisis	33
3.8 Penarikan Kesimpulan dan Saran	34
BAB 4 PERANCANGAN.....	35

4.1 Analisis Kebutuhan Sistem.....	35
4.1.1 Kebutuhan Fungsional.....	35
4.1.2 Kebutuhan Non-Fungsional	36
4.1.3 <i>Use Case Diagram</i>	37
4.1.4 <i>Use Case Scenario</i>	38
4.2 Pengumpulan Data	46
4.3 <i>Pre Processing Data</i>	47
4.3.1 <i>Data Selection</i>	47
4.3.2 <i>Data Cleansing</i>	49
4.3.3 <i>Data Transformation</i>	49
4.3.4 Pembagian Data	51
4.4 <i>Learning Process</i>	52
4.4.1 <i>Learning Process</i> Secara Manual.....	52
4.4.2 <i>Learning Process</i> Menggunakan <i>Weka</i>	60
4.5 Perancangan Sistem.....	62
4.5.1 <i>Sequence Diagram</i>	62
4.5.2 <i>Class Diagram</i>	68
4.6 Perancangan Visualisasi <i>Dashboard</i>	70
4.6.1 Perancangan Visualisasi Halaman <i>Login</i>	70
4.6.2 Perancangan Visualisasi Halaman <i>Dashboard</i>	71
4.6.3 Perancangan Visualisasi Halaman Prediksi Nasabah	72
4.6.4 Perancangan Visualisasi Halaman Lihat Akurasi	72
4.6.5 Perancangan Visualisasi Halaman Lihat <i>Tree</i>	73
4.6.6 Perancangan Visualisasi Halaman <i>Input Dataset</i>	73
4.6.7 Perancangan Visualisasi Halaman Data Pengajuan Nasabah	74
BAB 5 IMPLEMENTASI	76
5.1 Spesifikasi Sistem	76
5.1.1 Spesifikasi Perangkat Keras.....	76
5.1.2 Spesifikasi Perangkat Lunak	76
5.2 Batasan Implementasi	77
5.3 Implementasi Sistem	77
5.3.1 Implementasi Fungsi <i>Input Dataset</i>	77

5.3.2 Implementasi Fungsi Prediksi Nasabah	85
5.3.3 Implementasi Fungsi Lihat Akurasi	90
5.3.4 Implementasi Fungsi Lihat <i>Tree</i>	90
5.4 Implementasi Visualisasi	91
5.4.1 Implementasi Visualisasi Halaman <i>Login</i>	91
5.4.2 Implementasi Visualisasi Halaman <i>Dashboard</i>	91
5.4.3 Implementasi Visualisasi Halaman Prediksi Nasabah	93
5.4.4 Implementasi Visualisasi Halaman Lihat Akurasi	94
5.4.5 Implementasi Visualisasi Halaman Lihat <i>Tree</i>	95
5.4.6 Implementasi Visualisasi Halaman <i>Input Dataset</i>	96
5.4.7 Implementasi Visualisasi Halaman Data Nasabah	96
BAB 6 PENGUJIAN DAN ANALISIS	98
6.1 Evaluasi dan Validasi Algoritme <i>Data Mining</i>	98
6.1.1 <i>Confussion Matrix</i>	98
6.1.2 Kurva ROC	102
6.2 Pengujian <i>Black Box – Validation Testing</i>	103
6.2.1 Pengujian Fitur Memasukkan <i>Dataset</i>	103
6.2.2 Pengujian Fitur Melihat Data Pengajuan Nasabah	106
6.2.3 Pengujian Fitur Memasukkan Data Pengaju Pinjaman	108
6.2.4 Pengujian Fitur Melihat Jumlah Pengajuan	110
6.2.5 Pengujian Fitur Melihat Persentase Pengajuan	111
6.2.6 Pengujian Fitur Melihat <i>Time-Series</i> Pengajuan	113
6.2.7 Pengujian Fitur Melihat <i>Trend</i> Pengajuan	114
6.2.8 Pengujian Fitur Melihat Tingkat Akurasi	116
6.2.9 Pengujian Fitur Melihat Pohon Keputusan	117
6.2.10 Pengujian Fitur Autentifikasi	118
6.2.11 Pengujian Fitur Keluar	121
6.3 Pengujian <i>Black Box – Usability Testing</i>	122
BAB 7 PENUTUP	123
7.1 Kesimpulan	123
7.2 Saran	124
DAFTAR PUSTAKA	125

LAMPIRAN A WAWANCARA.....	129
LAMPIRAN B DIAGRAM PERANCANGAN SISTEM.....	132
LAMPIRAN C DATASET.....	134
LAMPIRAN D DATA YANG MENGANDUNG MISSING VALUES.....	137
LAMPIRAN E DATA HASIL <i>PRE PROCESSING DATA</i>	139
LAMPIRAN F DATA LATIH	141
LAMPIRAN G DATA UJI	143
LAMPIRAN H PERHITUNGAN <i>ENTROPY</i> DAN <i>GAIN</i>	144
LAMPIRAN I HASIL KUESIONER SUS	148



DAFTAR TABEL

Tabel 2. 1 Kajian Pustaka.....	7
Tabel 2. 2 Model <i>Confussion Matrix</i>	17
Tabel 2. 3 Simbol-Simbol Dalam Diagram <i>Use Case</i>	20
Tabel 2. 4 Simbol-Simbol Dalam Diagram <i>Activity</i>	21
Tabel 2. 5 Simbol-Simbol Dalam Diagram <i>Sequence</i>	22
Tabel 2. 6 Simbol-Simbol Dalam Diagram <i>Class</i>	22
Tabel 2. 7 Pertanyaan Kuesioner <i>System Usability Scale</i>	26
Tabel 4. 1 Kebutuhan Fungsional.....	35
Tabel 4. 2 Kebutuhan Non-Fungsional.....	36
Tabel 4. 3 Identifikasi <i>Use Case</i>	37
Tabel 4. 4 <i>Use Case Scenario</i> Memasukkan <i>Dataset</i>	38
Tabel 4. 5 <i>Use Case Scenario</i> Melihat Data Pengaju Nasabah.....	39
Tabel 4. 6 <i>Use Case Scenario</i> Memasukkan Data Pengaju Pinjaman	40
Tabel 4. 7 <i>Use Case Scenario</i> Melihat Hasil Prediksi.....	40
Tabel 4. 8 <i>Use Case Scenario</i> Melihat Jumlah Pengajuan.....	41
Tabel 4. 9 <i>Use Case Scenario</i> Melihat Persentase Pengajuan	42
Tabel 4. 10 <i>Use Case Scenario</i> Melihat <i>Time-Series</i> Pengajuan.....	42
Tabel 4. 11 <i>Use Case Scenario</i> Melihat <i>Trend</i> Pengajuan	43
Tabel 4. 12 <i>Use Case Scenario</i> Melihat Tingkat Akurasi	44
Tabel 4. 13 <i>Use Case Scenario</i> Melihat Pohon Keputusan.....	44
Tabel 4. 14 <i>Use Case Scenario</i> Autentifikasi	45
Tabel 4. 15 <i>Use Case Scenario</i> Keluar	46
Tabel 4. 16 Atribut <i>Dataset</i>	46
Tabel 4. 17 Perubahan Atribut <i>Dataset</i>	47
Tabel 4. 18 Potongan Data Hasil Pemilihan Atribut.....	48
Tabel 4. 19 Hasil <i>Data Transformation</i>	49
Tabel 4. 20 Potongan Data Hasil <i>Transformation Data</i>	50
Tabel 4. 21 Data untuk Perhitungan <i>Learning Process</i> Secara Manual	53
Tabel 4. 22 Perhitungan Nilai <i>Entropy Total</i>	56

Tabel 4. 23 Perhitungan Nilai <i>Entropy</i> dan <i>Gain</i> Tiap Atribut untuk Penentuan Node Akar.....	57
Tabel 4. 24 Perhitungan Nilai <i>Entropy</i> dan <i>Gain</i> Tiap Atribut untuk Penentuan Node Cabang 1.1	58
Tabel 5. 1 Spesifikasi Minimum Perangkat Keras	76
Tabel 5. 2 Spesifikasi Perangkat Lunak	76
Tabel 6. 1 Perhitungan <i>Confussion Matrix</i>	98
Tabel 6. 2 <i>Data Testing</i> Hasil Klasifikasi	99
Tabel 6. 3 <i>Data Testing</i> Hasil Klasifikasi pada <i>False Positives</i>	102
Tabel 6. 4 <i>Data Testing</i> Hasil Klasifikasi pada <i>False Negatives</i>	102
Tabel 6. 5 Kasus Uji Memasukkan <i>Dataset: Basic Flow</i>	103
Tabel 6. 6 Kasus Uji Memasukkan <i>Dataset: alt1</i>	105
Tabel 6. 7 Kasus Uji Melihat Data Nasabah: <i>Basic Flow</i>	106
Tabel 6. 8 Kasus Uji Memasukkan Data Pengaju Pinjaman: <i>Basic Flow</i>	108
Tabel 6. 9 Kasus Uji Memasukkan Data Pengaju Pinjaman: <i>alt1</i>	109
Tabel 6. 10 Kasus Uji Melihat Jumlah Pengajuan: <i>Basic Flow</i>	110
Tabel 6. 11 Kasus Uji Melihat Persentase Pengajuan: <i>Basic Flow</i>	112
Tabel 6. 12 Kasus Uji Melihat <i>Time-Series</i> Pengajuan: <i>Basic Flow</i>	113
Tabel 6. 13 Kasus Uji Melihat <i>Trend</i> Pengajuan: <i>Basic Flow</i>	115
Tabel 6. 14 Kasus Uji Melihat Tingkat Akurasi: <i>Basic Flow</i>	116
Tabel 6. 15 Kasus Uji Melihat Pohon Keputusan: <i>Basic Flow</i>	117
Tabel 6. 16 Kasus Uji Autentifikasi: <i>Basic Flow</i>	118
Tabel 6. 17 Kasus Uji Autentifikasi: <i>alt1</i>	119
Tabel 6. 18 Kasus Uji Keluar: <i>Basic Flow</i>	121
Tabel 6. 19 Hasil Pengisian Kuesioner oleh Pimpinan dan Admin KSP	122

DAFTAR GAMBAR

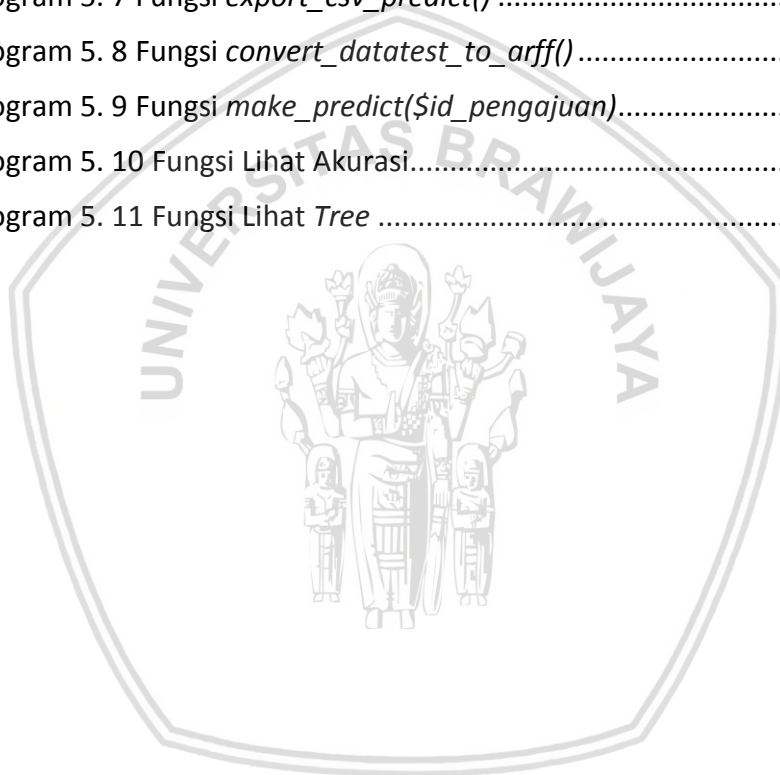
Gambar 2. 1 Tahapan KDD	13
Gambar 3. 1 Alur Penelitian	30
Gambar 4. 1 Proses Pembagian Data Menggunakan Fitur <i>Resample</i> pada <i>Weka</i>	52
Gambar 4. 2 Hasil <i>Tree</i> dari Perhitungan Pertama	58
Gambar 4. 3 Hasil <i>Tree</i> yang Terbentuk.....	60
Gambar 4. 4 Hasil <i>Tree</i> dengan Atribut JUMLAH_TANGGUNGAN adalah SEDIKIT Menggunakan <i>Weka</i>	60
Gambar 4. 5 Hasil <i>Tree</i> dengan Atribut JUMLAH_TANGGUNGAN adalah SEDANG dan BANYAK Menggunakan <i>Weka</i>	61
Gambar 4. 6 <i>Sequence Diagram</i> Melihat Data Pengajuan Nasabah.....	63
Gambar 4. 7 <i>Sequence Diagram</i> Memasukkan Data Pengaju Pinjaman	63
Gambar 4. 8 <i>Sequence Diagram</i> Melihat Hasil Prediksi.....	64
Gambar 4. 9 <i>Sequence Diagram</i> Melihat Jumlah Pengajuan.....	64
Gambar 4. 10 <i>Sequence Diagram</i> Melihat Persentase Pengajuan	65
Gambar 4. 11 <i>Sequence Diagram</i> Melihat <i>Time-Series</i> Pengajuan.....	65
Gambar 4. 12 <i>Sequence Diagram</i> Melihat <i>Trend</i> Pengajuan	66
Gambar 4. 13 <i>Sequence Diagram</i> Melihat Tingkat Akurasi	66
Gambar 4. 14 <i>Sequence Diagram</i> Melihat Pohon Keputusan	67
Gambar 4. 15 <i>Sequence Diagram</i> Autentifikasi	67
Gambar 4. 16 <i>Sequence Diagram</i> Keluar	68
Gambar 4. 17 <i>Class Diagram</i> sebagai <i>logical class</i>	68
Gambar 4. 18 <i>Class Diagram</i> sebagai <i>domain model</i>	69
Gambar 4. 19 <i>Physical Data Modelling</i>	69
Gambar 4. 22 Perancangan Visualisasi Halaman <i>Login</i>	70
Gambar 4. 23 Perancangan Visualisasi Halaman <i>Dashboard</i> Pimpinan KSP	71
Gambar 4. 24 Perancangan Visualisasi Halaman <i>Dashboard</i> Admin KSP	71
Gambar 4. 25 Perancangan Visualisasi Halaman Prediksi Nasabah	72
Gambar 4. 26 Perancangan Visualisasi Halaman Lihat Akurasi	72
Gambar 4. 27 Perancangan Visualisasi Halaman Lihat <i>Tree</i>	73
Gambar 4. 28 Perancangan Visualisasi Halaman <i>Input Dataset</i> Pimpinan KSP	73

Gambar 4. 29 Perancangan Visualisasi Halaman <i>Input Dataset</i> Admin KSP	74
Gambar 4. 30 Perancangan Visualisasi Halaman Lihat Data Pengajuan Nasabah Pimpinan KSP.....	74
Gambar 4. 31 Perancangan Visualisasi Halaman Lihat Data Pengajuan Nasabah Admin KSP	75
Gambar 5. 1 Implementasi Visualisasi Halaman <i>Login</i>	91
Gambar 5. 2 Implementasi Visualisasi Halaman <i>Dashboard</i> Pimpinan KSP	92
Gambar 5. 3 Implementasi Visualisasi Halaman <i>Dashboard</i> Pimpinan KSP (lanjutan).....	92
Gambar 5. 4 Implementasi Visualisasi Halaman <i>Dashboard</i> Admin KSP.....	93
Gambar 5. 5 Implementasi Visualisasi Halaman <i>Dashboard</i> Admin KSP (lanjutan)	93
Gambar 5. 6 Implementasi Visualisasi Halaman Prediksi Nasabah	94
Gambar 5. 7 Implementasi Visualisasi Halaman Hasil Prediksi Nasabah	94
Gambar 5. 8 Implementasi Visualisasi Halaman Lihat Akurasi	95
Gambar 5. 9 Implementasi Visualisasi Halaman Lihat <i>Tree</i>	95
Gambar 5. 10 Implementasi Visualisasi Halaman <i>Input Dataset</i> Pimpinan KSP... ..	96
Gambar 5. 11 Implementasi Visualisasi Halaman <i>Input Dataset</i> Admin KSP	96
Gambar 5. 12 Implementasi Visualisasi Halaman Data Nasabah Pimpinan KSP ..	97
Gambar 5. 13 Implementasi Visualisasi Halaman Data Nasabah Admin KSP	97
Gambar 6. 1 Hasil <i>Classifier C4.5 Weka</i>	99
Gambar 6. 2 Hasil <i>Threshold Curve Class Value</i> Macet.....	103
Gambar 6. 4 Hasil Pengujian Fitur Memasukkan <i>Dataset</i> Pimpinan KSP: <i>Basic Flow</i>	104
Gambar 6. 5 Hasil Pengujian Fitur Memasukkan <i>Dataset</i> Admin KSP: <i>Basic Flow</i>	105
Gambar 6. 6 Hasil Pengujian Fitur Memasukkan <i>Dataset</i> Pimpinan KSP: <i>alt1</i> ...	106
Gambar 6. 7 Hasil Pengujian Fitur Memasukkan <i>Dataset</i> Admin KSP: <i>alt1</i>	106
Gambar 6. 8 Hasil Pengujian Fitur Melihat Nasabah Pimpinan KSP: <i>Basic Flow</i>	107
Gambar 6. 9 Hasil Pengujian Fitur Melihat Nasabah Admin KSP: <i>Basic Flow</i>	108
Gambar 6. 10 Hasil Pengujian Fitur Memasukkan Data Pengaju Pinjaman: <i>Basic Flow</i>	109
Gambar 6. 11 Hasil Pengujian Fitur Memasukkan Data Pengaju Pinjaman: <i>alt1</i>	110

Gambar 6. 12 Hasil Pengujian Fitur Melihat Jumlah Pengajuan Pimpinan KSP: <i>Basic Flow</i>	111
Gambar 6. 13 Hasil Pengujian Fitur Melihat Jumlah Pengajuan Admin KSP: <i>Basic Flow</i>	111
Gambar 6. 14 Hasil Pengujian Fitur Melihat Persentase Pengajuan Pimpinan KSP: <i>Basic Flow</i>	112
Gambar 6. 15 Hasil Pengujian Fitur Melihat Persentase Pengajuan Admin KSP: <i>Basic Flow</i>	113
Gambar 6. 16 Hasil Pengujian Fitur Melihat <i>Time-Series</i> Pengajuan Pimpinan KSP: <i>Basic Flow</i>	114
Gambar 6. 17 Hasil Pengujian Fitur Melihat <i>Time-Series</i> Pengajuan Admin KSP: <i>Basic Flow</i>	114
Gambar 6. 18 Hasil Pengujian Fitur Melihat <i>Trend</i> Pengajuan Pimpinan KSP: <i>Basic Flow</i>	115
Gambar 6. 19 Hasil Pengujian Fitur Melihat <i>Trend</i> Pengajuan Admin KSP: <i>Basic Flow</i>	116
Gambar 6. 20 Hasil Pengujian Fitur Melihat Tingkat Akurasi: <i>Basic Flow</i>	117
Gambar 6. 21 Hasil Pengujian Fitur Melihat Pohon Keputusan: <i>Basic Flow</i>	118
Gambar 6. 22 Hasil Pengujian Fitur Autentifikasi Pimpinan KSP: <i>Basic Flow</i>	119
Gambar 6. 23 Hasil Pengujian Fitur Autentifikasi Admin KSP: <i>Basic Flow</i>	119
Gambar 6. 24 Hasil Pengujian Fitur Autentifikasi: <i>alt1</i>	120
Gambar 6. 25 Hasil Pengujian Fitur Autentifikasi: <i>Basic Flow</i>	121

DAFTAR KODE

Kode Program 5. 1 Fungsi <i>Input Dataset</i>	79
Kode Program 5. 2 Fungsi <i>export_csv()</i>	82
Kode Program 5. 3 Fungsi <i>get_data_from_data_train()</i>	84
Kode Program 5. 4 Fungsi <i>convert_datatrain_to_arff()</i>	85
Kode Program 5. 5 Fungsi <i>create_model()</i>	85
Kode Program 5. 6 Fungsi Prediksi Nasabah.....	88
Kode Program 5. 7 Fungsi <i>export_csv_predict()</i>	89
Kode Program 5. 8 Fungsi <i>convert_datatest_to_arff()</i>	89
Kode Program 5. 9 Fungsi <i>make_predict(\$id_pengajuan)</i>	90
Kode Program 5. 10 Fungsi Lihat Akurasi.....	90
Kode Program 5. 11 Fungsi Lihat <i>Tree</i>	90



DAFTAR LAMPIRAN

LAMPIRAN A WAWANCARA.....	129
A.1 Wawancara 1	129
A.2 Wawancara 2	131
LAMPIRAN B DIAGRAM PERANCANGAN SISTEM.....	132
B.1 Use Case Diagram	132
B.2 Sequence Diagram Memasukkan Dataset (SD-SRPK-1)	133
LAMPIRAN C DATASET.....	134
LAMPIRAN D DATA YANG MENGANDUNG MISSING VALUES.....	137
LAMPIRAN E DATA HASIL PRE PROCESSING DATA.....	139
LAMPIRAN F DATA LATIH	141
LAMPIRAN G DATA UJI	143
LAMPIRAN H PERHITUNGAN ENTROPY DAN GAIN	144
H.1 Perhitungan Untuk Penentuan Node Akar	144
H.2 Perhitungan Untuk Penentuan Node Cabang 1.1	146
LAMPIRAN I HASIL KUESIONER SUS	148
I.1 Kuesioner 1	148
I.2 Kuesioner 2	149



IQBAL TAUFIQ AHMAD NUR

CURRICULUM VITAE

INFORMATION SYSTEM LABORATORY GRADUATE

NOMOR INDUK MAHASISWA

145150400111009

TEMPAT, TANGGAL LAHIR

Kediri, 20 Oktober 1994

ALAMAT

Dusun Karangrejan RT 13 RW 7 Desa Tawang Kec. Wates
Kab. Kediri, 64174



RIWAYAT PENDIDIKAN

- 2001 - 2007 SD Negeri Tawang 1
- 2007 - 2010 SMP Negeri 1 Wates
- 2010 - 2013 SMK Negeri 1 Kediri
- 2014 - 2018 Universitas Brawijaya



PRESTASI PRIBADI

- 2015 Juara 2 Kategori Web Design Lomba Karya Sistem Komputer (BASIK III) Tingkat Universitas Brawijaya
- 2016 The Best of Expert Kategori Web Design IFEST 2.0: SMART AND FUN BRAWIJAYA Tingkat Universitas Brawijaya



PENGALAMAN PRIBADI

- 2015 Wakil Ketua Divisi Acara Jalan Sehat Dies Natalies PTIHK Ke-4
- 2016 Wakil Ketua Divisi Acara Seminar Build a Business Start Up Based on Socio-Friendly Technology
- 2016 Koordinator Divisi Acara Olimpiade FILKOM
- 2016 Anggota Divisi Acara Lomba i-FEST 3.0



BAHASA

Indonesia (Bahasa Pokok)
Inggris



PASSION

Sport
Music
Traveling



KONTAK

- +62 838 3357 5633
- iqbaltaufiq.an@gmail.com
- iqbaltaufiqan

BAB 1 PENDAHULUAN

1.1 Latar belakang

Dalam pembangunan nasional, lembaga finansial menjadi salah satu instrumen yang berperan penting di dalamnya. Lembaga finansial berperan dalam membantu meningkatkan taraf hidup masyarakat banyak dengan cara menyimpan dan meminjamkan dana kepada masyarakat (Hoggson, 1926). Salah satu wujud dari lembaga finansial di Indonesia adalah koperasi simpan pinjam.

Umumnya, koperasi simpan pinjam menghimpun dana dari masyarakat dan disalurkan kembali dalam bentuk kredit atau pinjaman. Kredit sendiri merupakan kemampuan untuk melaksanakan suatu pembelian atau mengadakan suatu pinjaman dengan suatu janji, pembayarannya akan dilakukan dan ditangguhkan pada suatu jangka waktu yang disepakati (Kohler, 1983). Proses pengajuan kredit di koperasi simpan pinjam sendiri terbilang sederhana dan tidak seperti pengajuan kredit di bank. Namun, diperlukan analisis yang mendalam untuk menentukan apakah pengajuan kredit tersebut bisa diterima atau tidak guna meminimalisir adanya risiko kredit macet yang secara langsung akan mengancam profitabilitas dan kemampuan koperasi simpan pinjam itu sendiri untuk bisa memberikan kredit selanjutnya. Ada beberapa macam preferensi dari nasabah yang wajib dianalisis seperti misal slip gaji, agunan, serta tanggungan nasabah pengaju pinjaman untuk menentukan macet tidaknya suatu pengajuan pinjaman pada KSP.

Hal tersebut juga peneliti temukan di Koperasi Simpan Pinjam Mitra Raya Wates. Berdasarkan hasil wawancara yang terlampir dalam Lampiran A.1, produk dari koperasi simpan pinjam ini adalah pinjaman. Proses penentuan pengajuan kredit dari nasabah dilakukan oleh pimpinan selaku pihak pengambil keputusan dan tidak menggunakan tenaga analis kredit mengingat hal tersebut tentu akan menambah pengeluaran perusahaan. Dalam proses pengambilan keputusan tersebut, pimpinan masih menggunakan pendekatan intuitif dan berdasarkan pengalaman yang sudah ada, mengingat *background* dari pimpinan bukanlah seorang analis kredit. Kondisi ini tentu akan menimbulkan permasalahan adanya kesalahan analisis kredit yang tentu saja dapat menyebabkan risiko kredit, seperti hilangnya nasabah, pembayaran dana pinjaman menjadi tidak pasti bahkan nasabah yang tidak mampu mengembalikan pinjaman (Menarianti, 2015). Selain itu, walaupun sudah dilakukan proses survei secara langsung kepada calon nasabah yang mengajukan pinjaman, namun hasil survei tidak bisa dijadikan tolak ukur utama oleh Pimpinan KSP mengingat fakta di lapangan menyebutkan bahwa hampir keseluruhan nasabah di KSP Mitra Raya Wates telah lolos survei dan layak untuk diberikan pinjaman, namun masih terdapat nasabah yang tidak sedikit mengalami kemacetan. Terdapat kurang lebih 100 nasabah yang mengalami kredit macet dari total kurang lebih 400 nasabah (Franico, 2018b). Terdapat juga data yang kian menumpuk dari tahun ke tahun namun tidak dimanfaatkan untuk pengambilan keputusan di KSP Mitra Raya Wates yang

menciptakan istilah yang biasa disebut “*Rich of Data but Poor of Information*” (Pramudiono, 2003). Padahal data yang bertumpuk-tumpuk tersebut sangat bisa dimanfaatkan oleh pihak KSP Mitra Raya Wates sebagai pendukung keputusan dalam hal pengajuan pinjaman guna mendeteksi kualitas kredit sejak dini.

Metode yang bisa digunakan untuk memanfaatkan tumpukan data adalah *data mining*. *Data mining* dapat dimanfaatkan untuk melakukan prediksi berdasarkan data masa lalu. *Data mining* sendiri adalah proses untuk menemukan pola yang berguna dan kecenderungan di dalam kumpulan data yang besar (Larose dan Daniel, 2005). Salah satu metode dari *data mining* yang mampu menangani data dalam jumlah besar adalah *decision tree* C4.5 yang merupakan algoritme pengklasifikasian paling populer dimana algoritme ini merupakan pengembangan dari algoritme ID3. Algoritme ini akan menghasilkan sebuah *rule* dalam bentuk pohon keputusan untuk memperoleh jawaban dari masalah yang dimasukkan dengan fitur penanganan terhadap tipe data numerik, *pruning tree*, dan penurunan *rule set*. Algoritme C4.5 ini juga menggunakan kriteria *gain* dalam menentukan fitur yang menjadi pemecah *node* pada pohon yang diinduksi (Prasetyo, 2014).

Sudah banyak penelitian terdahulu yang menggunakan metode C4.5 ini, penelitin pertama berjudul “Prediksi Kredit Macet Melalui Perilaku Nasabah pada Koperasi Simpan Pinjam dengan Menggunakan Metode Algoritme Klasifikasi C4.5” yang dilakukan oleh Sucipto (2015). Model algoritme C4.5 dalam penelitian ini diuji secara terukur menggunakan uji ROC/AUC dan juga *T-Test*. Hasil yang didapatkan secara keseluruhan adalah tingkat akurasi sebesar 91,06%, *precision* sebesar 100%, dan juga *recall* sebesar 78%. Sehingga secara garis besar penggunaan C4.5 pada penelitian ini masih dikatakan baik dan dapat digunakan sebagai pedoman untuk mendeteksi kredit macet sebelum dilakukan pengambilan keputusan diterima tidaknya pengajuan pinjaman nasabah tersebut. Penelitian kedua berjudul “*Implementation of Decision Tree Using C4.5 Algorithm in Decision Making of Loan Application by Debtor (Case Study: Bank Pasar of Yogyakarta Special Region)*” yang dilakukan oleh Amin *et. al.* (2015). Data yang digunakan dalam penelitian ini berjumlah 1000 data pinjaman Bank Pasar Daerah Istimewa Yogyakarta dari bulan Januari hingga Juli 2014. Untuk pengujian dilakukan pembagian data dengan skema 90% : 10%, 80% : 20%, 70% : 30%, dan 60% : 40%. Tingkat *precision* terbesar bernilai 78,08% dari skema pembagian data 90% : 10%. Nilai *recall* terbesar adalah 96,4% yang didapatkan dari skema pembagian data 80% : 20%, sedangkan skema pembagian data terbaik didapatkan dari pembagian data sebesar 80% : 20% dengan tingkat akurasi sebesar 74,5% dan *recall* sebesar 96,4%. Penelitian ketiga berjudul “*The Application of Decision Tree C4.5 Algorithm to Soil Quality Grade Forecasting Model*” yang dilakukan oleh Dongming *et. al.* (2016). Data tanah Lishu yang terletak di Provinsi Jilin dari tahun 2006 hingga 2008 dijadikan sebagai *dataset* pada penelitian ini. Terdapat enam atribut yang digunakan dalam penelitian ini, yaitu *Organic Matter*, Total N, Total P, Total K, AHN, dan *Class* sebagai atribut kelas yang merepresentasikan *black soil*, *chernozem*, *meadow soil*, *brown soil*, dan *sand soil* yang dibagi kedalam tingkatan A, B, C, D, dan E. Tingkat akurasi

yang didapatkan sistem sebesar 92,71%. Adapun hal yang membedakan penelitian ini dengan penelitian-penelitian terdahulu adalah penggunaan parameter yang berbeda dan juga dilakukan pengujian usability untuk menilai kualitas sistem yang dibuat kepada calon pengguna dari KSP Mitra Raya Wates.

Berdasarkan paparan di atas, perlu dilakukan penelitian tentang prediksi kredit macet berdasarkan preferensi nasabah yang diimplementasikan di KSP Mitra Raya Wates dengan menggunakan algoritme klasifikasi C4.5. Hasil dari prediksi akan ditampilkan dalam suatu *dashboard* berbasis *web* dengan menerapkan algoritme klasifikasi C4.5 menggunakan *tool Weka* yang diharapkan dapat mendukung pihak pemangku keputusan di KSP Mitra Raya Wates untuk memutuskan pengajuan pinjaman yang dilakukan oleh nasabah. Penelitian yang dilakukan berjudul **“Prediksi Kredit Macet Berdasarkan Preferensi Nasabah Menggunakan Metode Klasifikasi C4.5 pada Koperasi Simpan Pinjam Mitra Raya Wates”**.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dipaparkan di atas, maka terdapat beberapa rumusan masalah pada penelitian ini, diantaranya:

1. Bagaimana parameter yang menjadi preferensi nasabah untuk memprediksi kredit macet pada KSP Mitra Raya Wates?
2. Bagaimana algoritme C4.5 diimplementasikan dalam memprediksi kredit macet berdasarkan preferensi nasabah pada KSP Mitra Raya Wates?
3. Bagaimana tingkat akurasi dari implementasi algoritme C4.5 dalam memprediksi kredit macet berdasarkan preferensi nasabah pada KSP Mitra Raya Wates?
4. Bagaimana visualisasi *dashboard* yang memuat hasil prediksi kredit macet pada nasabah KSP Mitra Raya Wates?

1.3 Tujuan

Adapun tujuan yang ingin dicapai diantaranya:

1. Mengetahui parameter untuk memprediksi kredit macet berdasarkan preferensi nasabah pada KSP Mitra Raya Wates.
2. Mengimplementasikan algoritme C4.5 dalam memprediksi kredit macet berdasarkan preferensi nasabah pada KSP Mitra Raya Wates.
3. Menguji tingkat akurasi dari implementasi algoritme C4.5 dalam memprediksi kredit macet berdasarkan preferensi nasabah pada KSP Mitra Raya Wates.
4. Menghasilkan visualisasi *dashboard* yang memuat hasil prediksi kredit macet pada nasabah KSP Mitra Raya Wates.

1.4 Manfaat

Bagi Instansi:

1. Menyediakan rekomendasi yang dapat membantu pihak koperasi simpan pinjam untuk pengambilan keputusan dalam kasus pengajuan kredit dari nasabah.
2. Menambah parameter bagi pihak pengambil keputusan untuk menentukan apakah pengajuan kredit nasabah dapat diterima atau tidak.

Bagi Universitas Brawijaya:

1. Menambah referensi penelitian ilmiah yang pada waktu mendatang dapat digunakan sebagai dasar dalam melakukan penelitian-penelitian dengan topik yang sama.

1.5 Batasan masalah

Batasan dalam penelitian yang peneliti lakukan adalah sebagai berikut:

1. Data yang digunakan dalam penelitian ini adalah data yang didapatkan langsung dari KSP Mitra Raya Wates.
2. Metode yang digunakan dalam penelitian ini adalah metode klasifikasi dengan algoritme C4.5.
3. Pengujian tingkat akurasi menggunakan *Confussion Matrix* dan kurva ROC.
4. *Tools* yang digunakan untuk membantu penelitian ini adalah *Weka* dan *Microsoft Excel*.

1.6 Sistematika pembahasan

Sistematika dalam penyusunan penelitian ini adalah sebagai berikut:

BAB 1 : PENDAHULUAN

Pada bab ini menjelaskan tentang latar belakang masalah yang ada, rumusan masalah, tujuan, manfaat, batasan masalah serta sistematika penulisan yang digunakan.

BAB 2 : LANDASAN KEPUSTAKAAN

Bab ini menjelaskan mengenai dasar teori yang digunakan dalam mendukung topik penelitian. Pada bab ini juga memuat kajian pustaka yang terdapat referensi skripsi ataupun jurnal dengan penelitian yang hampir sama atau relevan.

BAB 3 : METODOLOGI PENELITIAN

Pada bab ini menjelaskan langkah – langkah yang ditempuh atau metode penelitian yang digunakan dalam melakukan penelitian ini. Metode tersebut memuat dalam pengambilan data, pengolahan data serta analisis dan hasil akhir atau bahkan rekomendasi yang diberikan dari hasil penelitian ini.

BAB 4 : PERANCANGAN

Bab ini memuat mengenai hal-hal terkait perancangan seperti *pre-processing* data, perancangan perangkat lunak, proses *learning*, dan perancangan *dashboard*.

BAB 5 : IMPLEMENTASI

Bab ini menjelaskan tentang realisasi dari hasil rancangan yang telah dibuat seperti pembuatan *dashboard* yang sesuai dengan kebutuhan pengguna serta mengimplementasikan *rule* dalam bentuk pohon keputusan ke dalamnya.

BAB 6 : PENGUJIAN DAN ANALISIS

Pada bab ini memuat hal-hal terkait hasil dari pengolahan data dan uji akurasi klasifikasi menggunakan algoritme C4.5.

BAB 7 : PENUTUP

Bab ini menjelaskan kesimpulan dari keseluruhan penelitian beserta saran yang diberikan berdasarkan hasil penelitian yang telah dilakukan.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Ada beberapa penelitian terdahulu yang menggunakan metode klasifikasi dengan algoritme *decision tree* C4.5, diantaranya adalah penelitian dengan judul “Prediksi Kredit Macet Melalui Perilaku Nasabah pada Koperasi Simpan Pinjam dengan Menggunakan Metode Algoritme Klasifikasi C4.5” yang dilakukan oleh Sucipto (2015). Model algoritme C4.5 dalam penelitian ini diuji secara terukur menggunakan uji ROC/AUC dan juga *T-Test*. Hasil yang didapatkan secara keseluruhan adalah tingkat akurasi sebesar 91,06%, *precision* sebesar 100%, dan juga *recall* sebesar 78%. Sehingga secara garis besar penggunaan C4.5 pada penelitian ini masih dikatakan baik dan dapat digunakan sebagai pedoman untuk mendeteksi kredit macet sebelum dilakukan pengambilan keputusan diterima tidaknya pengajuan pinjaman nasabah tersebut.

Penelitian selanjutnya adalah penelitian berjudul *Implementation of Decision Tree Using C4.5 Algorithm in Decision Making of Loan Application by Debtor (Case Study: Bank Pasar of Yogyakarta Special Region* (Amin et. al., 2015). Penelitian ini berfokus pada penggunaan algoritme C4.5 untuk menentukan diterima tidaknya pengajuan pinjaman pada Bank Pasar Daerah Istimewa Yogyakarta. Data berjumlah 1000 data yang didapatkan dari data pinjaman Bank Pasar Daerah Istimewa Yogyakarta dari bulan Januari hingga Juli 2014. Untuk pengujian dilakukan pembagian data sebesar 90% data latih dan 10% data uji, 80% data latih dan 20% data uji, 70% data latih dan 30% data uji, dan terakhir 60% data latih dan 40% data uji. Hasil akhir yang didapatkan adalah tingkat *precision* terbesar bernilai 78,08% yang didapatkan dari skema pembagian data 90%:10%. Nilai *recall* terbesar adalah 96,4% yang didapatkan dari pembagian data 80%:20%, sedangkan skema pembagian data terbaik didapatkan dari pembagian data sebesar 80%:20% dikarenakan pada pembagian data tersebut didapatkan tingkat akurasi dan *recall* terbaik dibandingkan dengan skema pembagian data lainnya. Tingkat akurasi yang didapatkan adalah sebesar 74,5% dan *recall* 96,4%.

Penelitian lainnya adalah penelitian dengan judul *The Application of Decision Tree C4.5 Algorithm to Soil Quality Grade Forecasting Model* (Dongming et. al., 2016). Penelitian ini berfokus pada proses pengklasifikasian untuk memprediksi tingkat kualitas tanah di Lishu. Dalam penelitian ini, data didapatkan dari proses pengestrakan data tanah Lishu yang terletak di provinsi Jilin dari kurun waktu 2006 hingga 2008. Terdapat enam atribut yang digunakan dalam penelitian ini, yaitu *Organic Matter*, Total N, Total P, Total K, AHN, dan *Class* sebagai atribut kelas yang merepresentasikan *black soil*, *chernozem*, *meadow soil*, *brown soil*, dan *sand soil* yang dibagi kedalam tingkatan A, B, C, D, dan E. Tingkat akurasi yang didapatkan setelah dilakukan pengujian dan verifikasi dari sistem yang telah dibuat adalah sebesar 92,71%.

Ringkasan penelitian terkait direpresentasikan pada Tabel 2.1:

Tabel 2. 1 Kajian Pustaka

No	Nama Peneliti	Tahun	Judul	Metode	Hasil
1.	Adi Sucipto	2015	Prediksi Kredit Macet Melalui Perilaku Nasabah pada Koperasi Simpan Pinjam dengan Menggunakan Metode Algoritme Klasifikasi C4.5	Implementasi <i>Data Mining</i> Menggunakan Algoritme C4.5	Tingkat akurasi sebesar 91,06%, <i>precision</i> sebesar 100%, dan juga <i>recall</i> sebesar 78%. Sehingga secara garis besar penggunaan algoritme C4.5 masih dikatakan baik dan dapat dijadikan salah satu pedoman untuk mendeteksi kredit macet sebelum ada keputusan diterima tidaknya pengajuan pinjaman nasabah tersebut.
2.	Rafik Khairul Amin, Indwiarti, dan Yuliant Sibaroni	2015	<i>Implementati on of Decision Tree Using C4.5 Algorithm in Decision Making of Loan Application by Debtor (Case Study: Bank Pasar of Yogyakarta Special Region</i>	Implementasi <i>Data Mining</i> Menggunakan Algoritme C4.5	Tingkat <i>precision</i> terbesar bernilai 78,08% yang didapatkan dari skema pembagian data 90%:10%. Nilai <i>recall</i> terbesar adalah 96,4% yang didapatkan dari pembagian data 80%:20%. Sedangkan skema pembagian data terbaik didapatkan dari pembagian data sebesar 80%:20% dikarenakan pada

No	Nama Peneliti	Tahun	Judul	Metode	Hasil
					pembagian data tersebut didapatkan tingkat akurasi dan <i>recall</i> terbaik dibandingkan dengan skema pembagian data lainnya. Tingkat akurasi yang didapatkan adalah sebesar 74,5% dan <i>recall</i> 96,4% dari penggunaan 1000 data yang didapatkan dari data pinjaman Bank Pasar Daerah Istimewa Yogyakarta dari bulan Januari hingga Juli 2014.
3.	Li Dongmin, Li Yan, Yuan Chao, Li Chaoran, Liu Huan, dan Zhan Lijuan	2016	<i>The Application of Decision Tree C4.5 Algorithm to Soil Quality Grade Forecasting Model</i>	Implementasi Data Mining Menggunakan Algoritme C4.5	Tingkat akurasi yang didapatkan setelah dilakukan pengujian dan verifikasi dari sistem yang telah dibuat adalah sebesar 92,71% dari penggunaan data tanah Lishu yang terletak di provinsi Jilin dari kurun waktu 2006 hingga 2008 dengan menggunakan 6 atribut diantaranya adalah <i>Organic</i>

No	Nama Peneliti	Tahun	Judul	Metode	Hasil
					<i>Matter</i> , Total N, Total P, Total K, AHN, dan Class sebagai atribut kelas yang merepresentasikan <i>black soil</i> , <i>chernozem</i> , <i>meadow soil</i> , <i>brown soil</i> , dan <i>sand soil</i> yang dibagi kedalam tingkatan A, B, C, D, dan E.
4.	Lokeh S. Katore, Bhakti S. Ratnaparkhi, dan Dr. Jayant S. Umale	2015	<i>Novel Professional Career Prediction and Recommendation Method for Individual Through Analytics on Personal Traits Using C4.5 Algorithm</i>	Implementasi <i>Data Mining</i> Menggunakan Algoritme C4.5	Tingkat akurasi yang dihasilkan sebesar 86% dari 200 data siswa dengan menggunakan 12 atribut yang berpengaruh terhadap karir seperti <i>team work</i> , <i>technical skill</i> , <i>honesty</i> , <i>decision making</i> , <i>leadership</i> , <i>discipline</i> , <i>adaptability</i> , <i>persuading</i> , <i>written and verbal communication</i> , dsb. Nilai dari setiap atribut tersebut merupakan penjumlahan dari jawaban siswa yang terukur dari skala 1 sampai 10.

No	Nama Peneliti	Tahun	Judul	Metode	Hasil
5.	Anik Andriani	2012	Penerapan Algoritme C4.5 pada Program Klasifikasi Mahasiswa Dropout	Implementasi <i>Data Mining</i> Menggunakan Algoritme C4.5	Tingkat akurasi yang dihasilkan sebesar 97,5% setelah divalidasi menggunakan <i>confussion matrix</i> . Sedangkan hasil evaluasi dan validasi dengan kurva ROC/AUC menghasilkan nilai di atas 0,9 yang dapat diklasifikasikan sebagai metode yang sangat baik.
6.	David Hartanto Kamagi dan Seng Hansun	2014	Implementasi Data Mining dengan Algoritme C4.5 untuk Memprediksi Tingkat Kelulusan Mahasiswa	Implementasi Data Mining Menggunakan Algoritme C4.5	Aplikasi <i>desktop</i> berhasil memprediksi kelulusan mahasiswa dengan persentase 87,.% dari 60 data latih dan 40 data uji yang digunakan.

Dari beberapa penelitian terdahulu yang telah diperlihatkan pada Tabel 2.1, maka dilakukan penelitian untuk memprediksi kualitas pengajuan kredit sejak dini berdasarkan preferensi nasabah dengan menggunakan metode C4.5 yang dianggap sebagai metode yang optimal dan akurat. Namun, terdapat beberapa hal yang membedakan penelitian yang dilakukan dengan penelitian-penelitian terdahulu seperti misal penggunaan atribut yang berbeda, dimana pada penelitian yang dilakukan tidak mengikutsertakan atribut jangka waktu pinjaman dan jasa pinjaman seperti halnya pada penelitian yang dilakukan oleh Sucipto (2015). Atribut jangka waktu pinjaman tidak dapat digunakan dikarenakan semua pengajuan kredit pada KSP Mitra Raya Wates berjangka waktu 7 bulan, sedangkan atribut jasa pinjaman tidak terdapat pada *database* yang disimpan oleh KSP Mitra Raya Wates. Selain itu, juga dilakukan pengujian *usability* guna mengetahui kualitas sistem yang dihasilkan, dimana pada keenam penelitian terdahulu tidak dilakukan pengujian *usability*. Pengujian dilakukan langsung kepada Pimpinan KSP dan Admin KSP sebagai pengguna sistem nantinya.

2.2 Profil KSP Mitra Raya

KSP Mitra Raya Wates merupakan salah satu lembaga finansial di daerah Kabupaten Kediri yang didirikan pada tanggal 18 Agustus 2011, dengan badan hukum nomor 518/BH/XVI.9/218/2011. Saat ini, KSP Mitra Raya Wates telah memiliki hampir 450 anggota yang tersebar baik untuk daerah Kabupaten Kediri hingga Kabupaten Blitar. KSP Mitra Raya Wates didirikan dengan tujuan untuk dapat membantu dan mendorong pertumbuhan perekonomian dalam rangka meningkatkan taraf hidup anggota.

Visi dari KSP Mitra Raya Wates sendiri yaitu menjadi Koperasi Simpan Pinjam pilihan masyarakat yang mandiri dan tangguh dengan berlandaskan kebersamaan dalam membangun ekonomi bersama yang berkeadilan secara berkelanjutan. Sedangkan misi dari Koperasi Simpan Pinjam ini adalah menyediakan jasa layanan simpan pinjam bagi anggota dengan cara:

1. Senantiasa menjaga dan meningkatkan kualitas pelayanan kepada para anggota dan calon anggota.
2. Meningkatkan pengetahuan, ketrampilan dan sikap kerja setiap insan.
3. Memberikan kesejahteraan bagi pengurus, pengelola harian serta seluruh anggotanya.

Dalam mencapai program yang telah tertera dalam visi dan misi KSP Mitra Raya, disediakan produk berupa pinjaman bagi anggota maupun calon anggota. Syarat untuk mendapatkan pinjaman ini adalah sebagai berikut:

1. Berstatus sebagai anggota koperasi atau calon anggota koperasi.
2. Mengisi formulir pinjaman.
3. Menyerahkan Foto Copy KTP suami istri jika sudah menikah.
4. Menyerahkan Foto Copy KK, slip gaji dan juga agunan.

Adapun proses pemberian kredit pada KSP Mitra Raya Wates adalah sebagai berikut:

1. Nasabah meminta dan mengisi blanko atau formulir pengajuan pinjaman beserta persyaratan pengajuan pinjaman kepada bagian *front desk* KSP Mitra Raya Wates.
2. *Front desk* akan mengecek formulir pengajuan pinjaman yang telah diisi oleh nasabah yang mengajukan pinjaman.
3. Setelah dilakukan pengecekan, selanjutnya formulir akan diajukan kepada Pimpinan KSP untuk persetujuan.
4. Sebelum disetujui, dilakukan survei kepada nasabah yang mengajukan pinjaman.
5. Apabila hasil survei baik, maka Pimpinan KSP akan memberikan formulir kepada Bendahara KSP untuk pencairan uang.

6. Uang diberikan kepada nasabah yang mengajukan pinjaman.

2.3 Kredit

Kredit berasal dari bahasa Yunani yaitu *cedere* yang berarti sebuah kepercayaan atau *credo* yang berarti saya percaya. Yang dimaksud dengan kredit adalah seseorang yang memperoleh kredit (debitur) maka mereka memperoleh kepercayaan. Sedangkan pemberi kredit (kreditur) adalah memberikan kepercayaan kepada seseorang yang diberikan pinjaman uang dan pasti diangsur setiap bulan (Kasmir, 2000).

Undang-undang perbankan juga memiliki pengertian lain yang dituangkan dalam undang-undang perbankan nomor 10 tahun 1998, dimana kredit merupakan penyediaan uang atau berupa tagihan yang dapat dipersamakan berdasarkan persetujuan uang atau sebuah kesepakatan pinjaman antara bank dengan pihak lain yang wajib melunasi utangnya dalam jangka waktu dengan diberikan sebuah bunga sebagai balas jasa karena sudah meminjamkan modal untuk peminjam.

Kriteria yang digunakan secara umum oleh lembaga finansial untuk pemberian kredit adalah sebagai berikut (Dendawijaya, 2005):

- a. *Character*

Kepribadian atau sifat dari debitur apakah benar-benar dapat dipercaya atau tidak.

- b. *Capital*

Besaran modal yang dimiliki oleh calon debitur.

- c. *Capacity*

Kemampuan calon debitur untuk melunasi kewajiban kredit.

- d. *Collateral*

Jaminan yang diserahkan kepada bank dari calon debitur.

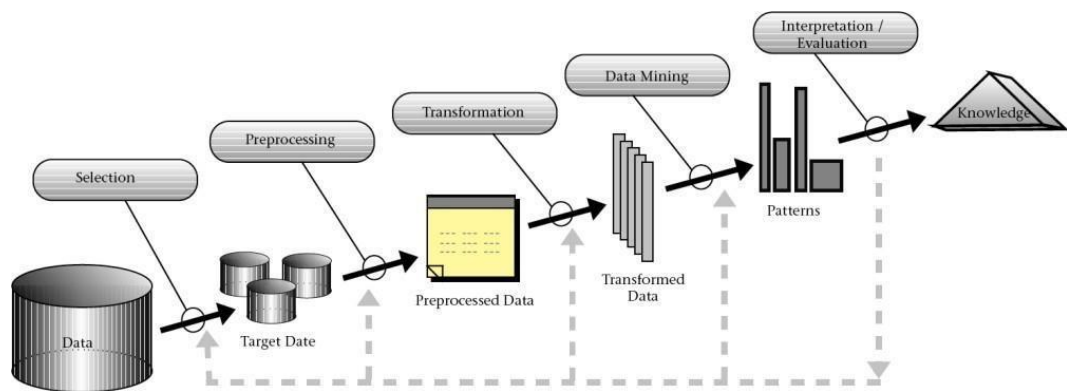
- e. *Condition of Economic*

Kondisi ekonomi dari calon debitur.

2.4 Knowledge Discovery in Database (KDD) dan Data Mining

2.4.1 Knowledge Discovery in Database (KDD)

Data mining dan *Knowledge Discovery in Database* (KDD) dapat digunakan untuk menggali informasi dari sekumpulan data dengan jumlah yang besar. Dua istilah tersebut saling terkait meskipun mempunyai arti yang berbeda. Dalam proses KDD, terdapat tahapan *data mining* (Han dan Kamber, 2006). Tahapan KDD dapat dilihat dalam Gambar 2.1:



Gambar 2. 1 Tahapan KDD

Sumber : Han dan Kamber (2006)

Tahapan KDD dapat dijabarkan sebagai berikut:

a. *Data Selection*

Dilakukan pemilihan data target untuk kemudian disimpan secara terpisah dengan data aslinya.

b. *Pre-Processing dan Cleaning Data*

Dilakukan proses penghapusan terhadap *inconsistent*, *incomplete*, atau data yang terduplikasi, serta memperbaiki data yang salah.

c. *Transformation*

Pada tahap ini, data digabungkan dengan cara agregasi.

d. *Data Mining*

Tahap ini berfokus pada penggalian sekumpulan data untuk menemukan suatu pola berdasarkan metode tertentu.

e. *Interpretation* atau Evaluasi

Pola yang telah dihasilkan dari proses *data mining*, kemudian akan diterjemahkan untuk kemudian dievaluasi dan divisualisasikan.

2.4.2 Data Mining

1. Pengertian *Data Mining*

Data mining merupakan kegiatan untuk menemukan suatu pola atau relasi dalam skala besar dengan menggunakan data historis yang telah dikumpulkan (Santosa, 2007).

Sedangkan menurut Larose dan Daniel (2005), *data mining* merupakan proses menemukan korelasi, pola, dan tren baru dengan menggunakan teknologi seperti teknis statistik dan matematis dari data berukuran besar yang telah disimpan sebelumnya.

2. Tujuan *Data Mining*

Data mining memiliki tiga tujuan, diantaranya (Baskoro, 2010):

- a. *Explanatory*, menjelaskan kegiatan penelitian.
- b. *Confirmatory*, mengonfirmasi hipotesis yang ada.
- c. *Exploratory*, melakukan analisis relasi pada data baru.

3. Pengelompokan *Data Mining*

Terdapat dua kategori utama dalam *data mining*, yaitu (Tan *et. al.*, 2005):

- a. *Descriptive mining*, yaitu proses dimana karakteristik data ditentukan. *Sequential mining*, *clustering* dan *association* termasuk dalam teknik ini.
- b. *Predictive mining*, yaitu penemuan pola dari penggunaan beberapa variabel lain yang melekat pada data. Salah satu tekniknya yaitu klasifikasi.

Sedangkan Larose dan Daniel (2005) mengelompokan *data mining* berdasarkan tugasnya, sebagai berikut:

- a. Deskripsi, penjelasan dari penggambaran pola data.
- b. Estimasi, hampir sama dengan klasifikasi, namun variabelnya bersifat numerik.
- c. Prediksi, hampir sama dengan klasifikasi dan estimasi, namun data yang dihasilkan akan terwujud pada masa mendatang.
- d. Klasifikasi, merupakan proses menemukan pola gambaran kelas data yang berbeda dan bertujuan menentukan kelas untuk suatu data (Han dan Kamber, 2006).
- e. *Clustering*, merupakan proses mengelompokkan objek ke dalam kelas objek yang memiliki karakteristik sama (Han dan Kamber, 2006).
- f. Asosiasi, menemukan kemunculan atribut dalam satu waktu.

2.5 Klasifikasi

Klasifikasi merupakan penemuan model (fungsi) yang mana kelas data atau konsep digambarkan dan dibedakan agar objek yang label kelasnya tidak diketahui dapat diprediksi (Han dan Kamber, 2006). Terdapat dua langkah proses dalam klasifikasi, yaitu *learning* (fase *training*), dimana *data training* dianalisis guna menghasilkan sebuah *rule* klasifikasi. Sedangkan yang kedua adalah klasifikasi dimana dalam proses ini menggunakan data uji guna menghasilkan akurasi dari *rule* klasifikasi (Han dan Kamber, 2006). Menurut Gorunescu (2011) proses klasifikasi didasarkan pada empat komponen, diantaranya:

1. Kelas, yaitu variabel terikat yang berupa kategorikal yang merepresentasikan label yang terdapat pada objek, seperti misal risiko penyakit jantung, risiko kredit, *customer loyalty*, jenis gempa.

2. *Predictor*, yaitu variabel bebas yang direpresentasikan oleh karakteristik (atribut) data, seperti misal: merokok, minum alkohol, tekanan darah, tabungan, aset, gaji.
3. *Training Dataset*, yaitu sekumpulan data yang berisi kelas dan *predictor* yang digunakan untuk menentukan kelas yang cocok berdasarkan *predictor*.
4. *Testing Dataset*, yaitu data baru yang akan diklasifikasikan oleh model yang telah dibuat dan akurasi klasifikasi dievaluasi.

2.6 Decision Tree

Decision tree merupakan pohon untuk mendapatkan jawaban dari masalah yang dimasukkan. Proses pada *decision tree* adalah pengubahan tabel menjadi model pohon, pengubahan model pohon menjadi *rule*, dan penyederhanaan *rule* (Basuki dan Syarif, 2004).

Decision tree merupakan metode klasifikasi yang paling sering digunakan. Hal ini dikarenakan proses pengerjaan yang sangat cepat dan hasil yang mudah dipahami (Oktaviana, 2016). Sudah banyak penelitian yang menggunakan *decision tree* guna mendapatkan hasil maksimal.

Terdapat tiga jenis *node* pada *decision tree*, yaitu (Utdirartatmo, 2005):

- a. *Root Node*, merupakan *node* paling atas, tidak ada *input*, dan memungkinkan *output* hanya satu.
- b. *Internal Node*, merupakan *node* percabangan, terdapat satu *input* dan *output* minimal dua.
- c. *Leaf Node* atau *Terminal Node*, merupakan *node* terakhir, hanya terdapat satu *input* dan tidak mempunyai *output*.

Kelebihan dari metode ini adalah:

- a. Hasil yang didapat mudah dipahami.
- b. Mudah dibangun dengan hanya memanfaatkan data yang lebih sedikit dibandingkan metode lain.
- c. Mampu mengolah data nominal dan kontinyu.
- d. Menggunakan teknik statistik sehingga dapat divalidasikan.
- e. Tingkat akurasi yang dihasilkan mampu menandingi metode lain. (Jadhav, et. al., 2016).

2.6.1 Algoritme C4.5

Algoritme C4.5 merupakan salah satu metode *decision tree* yang diperkenalkan pertama kali oleh Quinlan (1996). *Input* dari algoritme ini adalah *training samples* dan *samples*. *Training samples* berupa data contoh yang akan digunakan untuk membangun sebuah *tree* yang telah diuji kebenarannya.

Sedangkan *samples* merupakan *field-field* data yang nantinya akan digunakan sebagai parameter dalam melakukan klasifikasi data (Sunjana, 2010).

Algoritme C4.5 ini merupakan pengembangan dari algoritma ID3 (Larose, 2005). Perubahan yang membedakan algoritme C4.5 dengan ID3 adalah dapat menangani data bertipe numerik, melakukan *pruning tree*, dan penurunan (*deriving*) *rule set*. Algoritme C4.5 juga menggunakan kriteria *gain* dalam menentukan fitur yang menjadi pemecah *node* pada pohon yang diinduksi.

Secara umum algoritme C4.5 untuk membangun pohon keputusan adalah sebagai berikut:

1. Pilih atribut sebagai akar.
2. Buat cabang untuk masing-masing nilai.
3. Bagi kasus dalam cabang.
4. Ulangi proses untuk masing-masing cabang sampai semua kasus pada cabang memiliki kelas yang sama.

Untuk memilih atribut sebagai akar, didasarkan pada nilai *gain* tertinggi dari atribut-atribut yang ada. Untuk menghitung *gain* digunakan pada Persamaan 2.1 (Kusrini dan Luthfi, 2009):

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} * Entropy(S_i) \quad (2.1)$$

Sumber: (Kusrini dan Luthfi, 2009)

Dengan:

S : Himpunan kasus

A : Atribut

n : Jumlah partisi atribut A

|S_i| : Jumlah kasus pada partisi ke i

|S| : Jumlah kasus dalam S

Sehingga diperoleh nilai *gain* dari atribut yang paling tertinggi. Atribut dengan *information gain* tertinggi dipilih sebagai *test attribute* dari suatu *node*. Sedangkan perhitungan nilai *entropy* dapat dilihat pada Persaman 2.2 (Kusrini dan Luthfi, 2009):

$$Entropy(S) = \sum_{i=1}^n - p_i * \log_2 p_i \quad (2.2)$$

Sumber: (Kusrini dan Luthfi, 2009)

Dimana:

S : Himpunan kasus

n : Jumlah partisi S

p_i : Proporsi dari S_i terhadap S

2.6.2 Pruning dalam Decision Tree

Dalam pohon keputusan, dikenal istilah *pruning* yaitu pemangkasan cabang yang tidak terlalu besar pengaruhnya agar diagram dihasilkan lebih akurat dan *simple*.

Ada dua pendekatan *pruning* yang digunakan:

- a. *Prepruning*, menghentikan proses pembentukan cabang lebih awal pada titik tertentu. Pada kasus data yang beragam, kompleksitas semakin besar jika perulangan pembuatan cabang yang diperbolehkan juga semakin besar. Kekurangan lainnya, diagram pohon yang dihasilkan kurang akurat jika jumlah perulangan terlalu kecil.
- b. *Postpruning*, proses pemotongan cabang dilakukan di akhir, dalam artian setelah sebuah pohon keputusan terbentuk.

Algoritme C4.5 menggunakan *pessimistic pruning*, dimana dilakukan pengkalkulasian tingkat *error* yang nantinya digunakan sebagai acuan dalam pemangkasan cabang pohon keputusan. Baik *postpruning* dan *prepruning* dapat dikombinasikan. Walaupun pohon keputusan yang muncul setelah *pruning* akan lebih singkat, namun terkadang masih muncul repetisi dan replikasi cabang.

2.7 Metode Evaluasi dan Validasi Algoritme *Data Mining*

2.7.1 *Confussion Matrix*

Confussion Matrix merupakan metode yang memanfaatkan *table* matriks seperti pada Tabel 2.2. dimana tiap kolom pada matriks adalah contoh kelas prediksi, sedangkan tiap baris mewakili kejadian di kelas sebenarnya (Gorunescu, 2011).

Tabel 2. 2 Model *Confussion Matrix*

Klasifikasi yang benar	Diklasifikasikan sebagai	
	+	-
+	<i>True Positives</i>	<i>False Negatives</i>
-	<i>False Positives</i>	<i>True Negatives</i>

Sumber: Bramer (2007)

Keterangan tabel:

1. *True positives* merupakan jumlah *record* positif yang diklasifikasikan sebagai positif,
2. *False positives* merupakan jumlah *record* negatif yang diklasifikasikan sebagai positif,
3. *False negatives* merupakan jumlah *record* positif yang diklasifikasikan sebagai negatif,
4. *True negatives* merupakan jumlah *record* negatif yang diklasifikasikan sebagai negatif.

Confussion matrix merepresentasikan tingkat akurasi dari proses klasifikasi yang telah dilakukan. Tingkat akurasi menunjukkan proporsi jumlah prediksi benar. Perhitungan akurasi dapat dilihat pada Persamaan 2.3.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.3)$$

Sumber: (Maurina, 2015)

2.7.2 Kurva ROC

Kurva ROC menunjukkan akurasi dan membandingkan klasifikasi secara visual. ROC mengekspresikan *confussion matrix*. ROC adalah grafik dua dimensi dengan *false positives* sebagai garis horizontal dan *true positives* sebagai garis vertikal. *The area under the curve* (AUC) dihitung untuk mengukur perbedaan performansi metode yang digunakan.

Untuk klasifikasi data mining, nilai AUC dapat dibagi menjadi beberapa kelompok (Gorunescu, 2011):

- a. 0.90 – 1.00 = Klasifikasi sangat baik
- b. 0.80 – 0.90 = Klasifikasi baik
- c. 0.70 – 0.80 = Klasifikasi cukup
- d. 0.60 – 0.70 = Klasifikasi buruk
- e. 0.50 – 0.60 = Klasifikasi salah

2.8 Weka

Weka merupakan aplikasi *data mining open source* yang berbasis *java*. Aplikasi ini dikembangkan pertama kali pada tahun 1994 oleh Universitas Waikato di Selandia Baru sebelum menjadi bagian dari Pentaho. Aplikasi *weka* terdiri dari koleksi algoritme *machine learning* yang dapat digunakan untuk melakukan generalisasi atau formulasi dari sekumpulan *data sampling*. Inti dari kelebihan pada aplikasi *weka* terletak pada algoritme yang makin lengkap dan canggih, namun letak keberhasilan *data mining* tetap ditentukan oleh manusia itu sendiri sebagai interpreter. Keberhasilan *data mining* itu berdasarkan pada pengumpulan data yang berkualitas tinggi, penggunaan model dan algoritme yang tepat (Susanto, 2012).

2.9 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem adalah proses dibangunnya layanan atau produk sesuai dengan kebutuhan pengguna beserta dengan batasan-batasan yang berlaku pada saat produk digunakan atau dikembangkan. Arti kata kebutuhan mengacu pada standar IEEE nomor: *IEEE-STD-1220-1998* merupakan proses pengidentifikasian produk atau proses secara operasional maupun fungsional menggunakan pernyataan yang tidak ambigu, dapat diuji, dan terukur agar produk dapat diterima oleh konsumen.

Kebutuhan pengguna secara umum terbagi menjadi dua kategori utama, yaitu kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan fungsional merepresentasikan hal apa saja yang dapat dilakukan oleh sistem, sedangkan kebutuhan non-fungsional merepresentasikan batasan dalam hal batasan kualitas dan batasan *usability* terhadap sistem.

Hasil dari proses analisis kebutuhan sistem dapat digunakan sebagai bentuk kesepakatan terhadap kebutuhan yang harus dipenuhi antara pengembang dengan pengguna, sebagai dasar untuk proses perancangan sistem, sebagai dasar untuk melakukan validasi terhadap seluruh kebutuhan agar terspesifikasi dengan benar, lengkap, dan konsisten.

2.10 Perancangan Sistem

Menurut Pressman (2010), perancangan perangkat lunak adalah proses penerjemahan hasil dari analisis kebutuhan menjadi *blueprint* untuk pembangunan perangkat lunak. *Blueprint* sendiri merupakan penggambaran desain pada abstraksi tingkat tinggi yang dapat langsung ditelurusi pada sistem. Perbaikan berikutnya mengacu pada desain dengan tingkat abstraksi yang jauh lebih rendah.

Berikut ini merupakan prinsip dalam melakukan perancangan (Pressman, 2010):

- Hasil rancangan harus dapat dilacak dalam model persyaratan.
- Arsitektur sistem yang akan dibangun selalu dipertimbangkan.
- Rancangan dari data dan rancangan fungsi memiliki urgensi yang sama.
- Harus hati – hati dalam merancang antarmuka.
- Desain antarmuka pengguna harus sesuai dengan kebutuhan pengguna dan berfokus pada kemudahan pengguna.
- Komponen harus memiliki keterikatan yang rendah antara satu sama lain dan dengan lingkungan eksternal.
- Model rancangan harus mudah dimengerti.
- Rancangan harus dikembangkan secara iteratif.

Sebelum dilanjutkan ke tahap selanjutnya, pada tahap ini tim pengembang dapat menilai kualitas hasil rancangan dengan cara menganalisis apakah terdapat kesalahan, inkonsistensi, kelalaian, pencarian alternatif yang lebih baik, dan juga menganalisis apakah model rancangan akan sesuai dengan jadwal, biaya, dan batasan yang telah ditetapkan.

McGlaughlin dalam Pressman (2010) menyarankan tiga karakteristik yang berfungsi sebagai panduan untuk evaluasi perancangan yang baik, yaitu:

1. Semua persyaratan eksplisit dan implisit harus terimplementasi dan juga terakomodasi dalam sebuah rancangan.
2. Sebuah rancangan harus mudah dibaca dan dipahami oleh *developer*, *tester*, dan mereka yang akan mendukung perangkat lunak.

Sebuah rancangan harus menyediakan gambaran lengkap dari perangkat lunak, data, fungsional, dan domain perilaku dari perspektif implementasi.

2.11 UML (Unified Modelling Language)



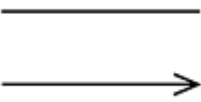
UML (Unified Modelling Language) merupakan standar bahasa yang digunakan untuk menspesifikasikan, memvisualisasikan, mengonstruksikan, dan mendokumentasikan artefak dari perangkat lunak yang akan dikembangkan (Pressman, 2010). UML memiliki diagram-diagram yang digunakan untuk mendesain suatu sistem atau perangkat lunak, diantaranya yaitu:

1. *Use Case Diagram*
2. *Activity Diagram*
3. *Sequence Diagram*
4. *Class Diagram*

2.11.1 Use Case Diagram

Use Case menjelaskan urutan kegiatan yang dilakukan aktor dan sistem untuk mencapai suatu tujuan tertentu dalam *Use case Specification*. Namun, *use case* hanya menjelaskan tentang apa saja yang dilakukan oleh aktor dan sistem dan tidak menjelaskan bagaimana aktor dan sistem melakukan kegiatan tersebut. *Use case specification* terdiri dari nama *use case*, deskripsi singkat (*Brief Description*), Aliran Normal (*Basic flow*), Aliran Alternatif (*Alternate Flow*), *Special Requirement*, *Pre-Condition* dan *Post-Condition* (Pressman, 2010). Simbol-simbol yang digunakan pada diagram *use case* dapat dilihat pada Tabel 2.3.

Tabel 2. 3 Simbol-Simbol Dalam Diagram *Use Case*




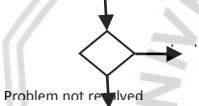
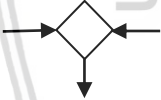



No	Simbol	Deskripsi
1	 An Actor	Actors. Mendefinisikan suatu peran yang mana pengguna dapat berinteraksi dengan sistem. Pengguna dapat berupa manusia ataupun sistem yang lain
2	 A Use Case	Use Case. Mendeskripsikan bagaimana aktor menggunakan sistem untuk mencapai tujuan dan apa yang sistem lakukan untuk aktor untuk mendapatkan tujuannya.
3		Communicates association. Aktor dan <i>use case</i> yang berinteraksi akan terhubung melalui simbol ini. Simbol asosiasi dengan panah mengindikasikan elemen mana yang memulai interaksi.

Sumber: Kurt Bittner & Ian Spence (2003)

2.11.2 Activity Diagram

Activity Diagram merupakan representasi grafis berupa aliran dari interaksi berdasarkan scenario spesifik (Pressman 2010). Simbol-simbol yang digunakan pada diagram *activity* dapat dilihat pada Tabel 2.4.

Tabel 2. 4 Simbol-Simbol Dalam Diagram Activity




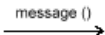
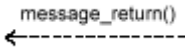


No	Simbol	Deskripsi
1		Initial node. Mendefinisikan titik mulai suatu proses.
2		Actions. Mendefinisikan notasi yang menggambarkan langkah-langkah yang terjadi.
3		Flow. Merupakan panah dalam diagram yang mengindikasikan alur antar- <i>actions</i> .
4		Decisions. Merupakan kondisi tertentu yang menentukan alur keluar.
5		Merge. Merupakan penggabungan alur yang sebelumnya terpisah oleh <i>decisions</i> .
6		Fork. Merupakan <i>bar</i> hitam untuk percabangan aktivitas yang secara bersamaan.
7		Join. Untuk menyatukan lagi alur yang terpisah oleh <i>fork</i> .
8		Activity Final. Merupakan titik akhir proses.

Sumber: Booch, et. al. (2005)

2.11.3 Sequence Diagram

Diagram sekuen digunakan untuk menggambarkan *dynamic view* dari sistem (Booch, et al., 2005). Simbol-simbol yang digunakan pada diagram *Sequence* ditunjukkan pada Tabel 2.5.

Tabel 2. 5 Simbol-Symbol Dalam Diagram *Sequence*

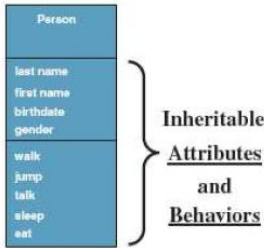
No	Simbol	Deskripsi
1		Entity. Mendefinisikan suatu entitas yang mempunyai atribut yang memiliki data yang bisa direkam.
2		Boundary. Mendefinisikan suatu <i>interface</i> yang menghubungkan pengguna dengan sistem.
3		Control. Mendefinisikan kontrol atas aktivitas yang dilakukan oleh sebuah kegiatan.
4		Message. Mendefinisikan pengiriman pesan.
5		Return Values. Mendefinisikan hasil dari pengiriman pesan.
6		Lifelines. Mengindikasikan urutan dari kehidupan.
7		Bar. Menunjukkan periode waktu ketika aktif dalam interaksi.

Sumber: Booch, *et.al* (2005)

2.11.4 Class Diagram

Class diagram atau diagram kelas digunakan untuk memodelkan tampilan statis perancangan dari sebuah sistem. Diagram kelas tidak hanya penting untuk memvisualisasikan, menentukan, dan mendokumentasikan model struktural, tapi juga penting untuk menyusun sistem yang dapat dijalankan melalui teknik *forward* maupun *reverse*. Diagram kelas secara umum memuat tiga hal, yaitu kelas, *interfaces*, dan hubungan dependensi, generalisasi, serta asosiasi (Booch, et al., 2005). Pada Tabel 2.6 menjelaskan tentang simbol-simbol yang digunakan dalam pembuatan *class diagram* pada umumnya.

Tabel 2. 6 Simbol-Symbol Dalam Diagram *Class*

No	Simbol	Deskripsi
1		<p>Attribute. Mendefinisikan sekumpulan data yang dimiliki oleh objek.</p> <p>Behavior. Kumpulan dari sesuatu yang dapat dilakukan oleh objek dan terkait dengan fungsi-fungsi yang bertindak pada data objek (atribut). Pada siklus berorientasi objek, perilaku objek merujuk kepada metode, operasi, atau fungsi.</p>

No	Simbol	Deskripsi
2		Inheritance. Menunjukkan bahwa satu kelas merupakan turunan dari kelas lain.
3		Association. Menunjukkan bahwa objek dari satu kelas berhubungan dengan kelas lain.
4		Aggregation. Menunjukkan bahwa contoh objek dari satu kelas terdiri dari contoh objek dari kelas lain.
5		Composition. Menunjukkan hubungan dimana satu kelas bertanggung jawab atas pembuatan dan perusakan bagian-bagian dalam kelas lainnya. Jika satu kelas rusak, maka kelas lain juga rusak.

Sumber: Booch, *et.al* (2005)

2.12 Implementasi Sistem

Implementasi sistem merupakan proses membangun sebuah sistem yang dapat berfungsi secara utuh dengan cara menerjemahkan rancangan yang telah dibuat kedalam bentuk bahasa pemrograman tertentu. Tahap implementasi termasuk dalam tahapan siklus pengembangan sistem.

2.12.1 Object Oriented Programming (OOP)

Object Oriented Programming merupakan salah satu paradigma dalam pembangunan sistem yang memandang segala sesuatu merupakan objek dan mempunyai atribut yang melekat, serta memiliki *behavior* (Shalahudin & S, 2015). Keuntungan mengembangkan sistem dengan menggunakan pemrograman berorientasi objek yaitu:

1. Meningkatkan produktivitas karena objek yang ditemukan dapat dipakai ulang (*reusable*).
2. Meningkatkan kecepatan dalam mengembangkan sistem karena telah dirancang dengan baik dan benar sehingga kesalahan dalam implementasi dapat diminimalisir.
3. Mudah dalam pemeliharaan karena menggunakan objek.
4. Adanya konsistensi dengan menggunakan konsep pewarisan (*inheritance*).
5. Meningkatkan kualitas perangkat lunak karena pengembangan didasari atas kejadian nyata, sehingga kebutuhan pengguna terpenuhi.

2.12.2 Hyper Text Markup Language (HTML)

Hyper Text Markup Language atau disingkat HTML adalah bahasa pemrograman yang digunakan dalam web sebagai bahasa untuk pertukaran dokumen web (Sibero, 2011).

Hypertext adalah teks yang membawa konten halaman web dan informasi pemrograman yang dibutuhkan untuk menampilkan halaman tersebut dan menghubungkannya ke halaman lain. Hypertext dibentuk dengan *markup language*, yaitu sebuah kumpulan symbol dan kode terstandar yang dapat diinterpretasikan oleh semua *browser*.

Markup digunakan untuk menyampaikan dua macam informasi tentang teks atau konten lainnya dalam halaman web, yaitu mengidentifikasi struktur yang dibutuhkan konten dan menentukan penyajian elemen-elemen yang ada.

2.12.3 Cascading Style Sheet (CSS)

Cascading Style Sheet (CSS) merupakan salah satu bahasa pemrograman web untuk mengendalikan beberapa komponen dalam sebuah web, sehingga akan lebih terstruktur dan seragam (Riyanto, 2009).

CSS menggunakan sebuah aturan *style* untuk menentukan penyajian visual dokumen HTML. Aturan *style* ini diintegrasikan dengan halaman web dengan berbagai cara, antara lain:

- *Internal style*: CSS disimpan di area <head> pada suatu halaman HTML dan *style* hanya digunakan untuk halaman tersebut.
- *External style*: CSS disimpan di file terpisah yang berformat .css dan *style* dapat digunakan untuk beberapa halaman web.
- *Inline style*: *Style* hanya diterapkan pada elemen tertentu dalam sebuah dokumen.

2.12.4 Hypertext Preprocessor (PHP)

PHP adalah singkatan dari *Hypertext Preprocessor*, yaitu Bahasa pemrograman *web server-side* yang dapat digunakan secara gratis. PHP merupakan *script* yang digunakan bersamaan dengan HTML yang berada pada server (*server side HTML embedded scripting*). PHP digunakan untuk membuat halaman *website* yang dinamis yang artinya halaman dibuat pada saat diminta oleh klien. Mekanisme ini menyebabkan informasi yang diterima klien selalu *up to date* (Anhar, 2010).

Beberapa kelebihan PHP, antara lain:

- Sederhana, PHP sangat mudah dipelajari karena memiliki banyak fungsi *built in* yang dapat menangani kebutuhan standar pembuatan aplikasi web.

- Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang mudah dipahami memiliki banyak referensi.
- PHP adalah bahasa open source yang dapat digunakan di berbagai sistem operasi. Selain itu, PHP dapat dijalankan secara runtime melalui *console* atau menjalankan perintah-perintah sistem. *Source code* yang tersedia secara gratis memungkinkan para *developer* dalam melakukan perbaikan, pengembangan, dan menemukan bug dalam Bahasa PHP.
- *Web server* yang mendukung PHP sangat banyak, seperti Apache, IIS, Lighttpd, dan Xitami dengan konfigurasi yang mudah.
- PHP juga dilengkapi dengan berbagai macam pendukung lain seperti dukungan langsung untuk beberapa *database*, seperti Oracle, PostgreSQL, MySQL, dan lain-lain.

2.12.5 Codeigniter

Codeigniter adalah aplikasi *open source* berupa *framework* dengan model MVC (*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan PHP (Supono dan Putratama, 2016). *Codeigniter* memudahkan pengembang web untuk membuat aplikasi web dengan cepat dan mudah dibandingkan dengan membuat dari awal.

Kelebihan *Codeigniter* (CI) dibandingkan dengan *framework* lain:

- Performa yang cepat: *Codeigniter* merupakan *framework* yang paling cepat dibanding *framework* lain.
- Konfigurasi yang sedikit (*nearly zero configuration*): Untuk menggunakan *codeigniter* dengan pengaturan yang standar, hanya perlu mengubah sedikit *file* pada folder *config*.
- Menggunakan konsep MVC: Pengerjaan antara logika dengan *layout* telah dipisahkan, sehingga *programmer* dan *designer* dapat mengerjakan tugas masing-masing dengan mudah.
- Banyak komunitas: Banyaknya komunitas CI memudahkan *developer* berdiskusi satu sama lain.
- Dokumentasi lengkap: Setiap paket instalasi *codeigniter* sudah disertai *user guide* yang lengkap dan mudah dipahami.

2.13 Black-Box Testing

Black-Box Testing merupakan teknik pengujian dimana kebutuhan fungsional sistem dilakukan pengecekan guna mengetahui kesalahan *error* pada perangkat lunak. Pengujian *blackbox* memiliki beberapa kelebihan sebagai berikut:

1. Pengujian dilakukan dari sudut pandang pengguna dan akan membantu dalam mengungkap ketidaksesuaian dalam spesifikasi perangkat lunak.

2. *Tester* tidak perlu tahu bahasa pemrograman atau bagaimana perangkat lunak diimplementasikan.
3. Pengujian dapat dilakukan oleh badan independen dari pengembang, memungkinkan untuk perspektif objektif.
4. *Test case* dapat langsung dirancang setelah spesifikasi telah lengkap.

2.13.1 Validation Testing

Validation Testing merupakan teknik pengujian dalam lingkup metode *Black Box* yang mana kesesuaian di uji. Pengujian ini dilakukan untuk mengetahui apakah sistem yang dibangun berjalan sesuai dengan kebutuhan yang telah didefinisikan atau tidak. Pengujian ini termasuk dalam *black-box testing*, karena pengujian dilakukan tanpa menunjukkan proses berjalannya sebuah fungsi yang sebenarnya, tetapi pengujian validasi melihat hasil yang dikeluarkan apakah sesuai dengan kebutuhan atau tidak, dan lebih ditekankan untuk menemukan konformitas antara kinerja sistem dengan daftar spesifikasi kebutuhan (Indriati, 2010).

2.13.2 Usability Testing

Usability merupakan analisis kualitatif untuk menentukan seberapa mudah *user* menggunakan antarmuka suatu aplikasi (Nielsen, 2012). *Usability Testing* merupakan teknik pengujian dalam lingkup metode *Black Box* untuk mengukur efisiensi, kemudahan dalam mempelajari, dan kemampuan untuk mengingat bagaimana berinteraksi tanpa mengalami kesulitan (Bauer, 2010). *Usability Testing* pada umumnya menggunakan teknik pertanyaan bagi *user* dalam bentuk kuesioner, *field observation*, dan *System Usability Scale* (SUS) untuk mendapatkan informasi tentang kualitas dari sistem yang dibuat.

2.13.2.1 System Usability Scale (SUS)

Salah satu metode untuk mengukur tingkat *usability* adalah dengan menggunakan *System Usability Scale* (SUS). Metode ini dapat digunakan pada *sample* yang jumlahnya sedikit, jumlah *sample* bisa dua *user* (Sauro, 2011). Dalam metode SUS ini menggunakan 10 buah pertanyaan. (Sauro, 2013). Adapun pertanyaan kuesioner yang digunakan dalam metode SUS dapat dilihat pada Tabel 2.7.

Tabel 2. 7 Pertanyaan Kuesioner System Usability Scale

No	Pertanyaan
1	Saya berfikir bahwa saya akan lebih sering menggunakan aplikasi ini
2	Saya menemukan sistem tidak harus dibangun serumit ini
3	Saya berfikir aplikasi ini mudah digunakan
4	Saya berfikir bahwa saya akan membutuhkan bantuan orang teknis untuk dapat menggunakan aplikasi ini

No	Pertanyaan
5	Saya menemukan berbagai fungsi pada aplikasi ini diintegrasikan dengan baik
6	Saya berfikir terdapat terlalu banyak ketidaksesuaian pada sistem ini
7	Saya bayangkan bahwa kebanyakan orang akan mudah untuk mempelajari aplikasi ini dengan cepat
8	Saya menemukan aplikasi ini sangat rumit untuk digunakan
9	Saya merasa sangat percaya diri untuk menggunakan aplikasi ini
10	Saya perlu belajar banyak hal sebelum saya bisa memulai menggunakan aplikasi ini

Untuk melakukan perhitungan nilai SUS dapat dilakukan dengan cara memberi bobot antara 0 sampai dengan 4 dengan aturan sebagai berikut (Sauro, 2011):

1. Item dengan nomor ganjil merupakan pernyataan positif yang nilainya didapat dengan cara posisi skala – 1.
2. Item dengan nomor genap merupakan pernyataan negatif yang nilainya didapat dengan cara 5 – posisi skala.

Hasil yang didapat dari total pembobotan kemudian dikalikan dengan 2.5, kemudian nilai SUS dirata-rata sesuai dengan rumus pada Persamaan 2.4.

$$\text{nilai rata - rata} = \sum_{i=1}^n \frac{x_i}{N} \quad (2.4)$$

Sumber: (Soleh, 2005)

Dimana:

Xi = Nilai responden

N = Jumlah responden

Berdasarkan hasil rata-rata, nilainya dapat dikelompokkan ke dalam tiga kategori dengan rentang yang berbeda-beda yaitu:

1. *Not Acceptable* : 0 – 50
2. *Marginal* : 50 – 70
3. *Acceptable* : 70 - 100

2.14 Konsep Dasar *Dashboard*

2.14.1 Definisi *Dashboard*

Enterprise dashboard merupakan antar muka yang memberikan visualisasi berupa bagan, *report*, indikator visual, dan mekanisme peringatan secara dinamis dalam pengelolaan inisiatif bisnis (Malik, 2005).

Information dashboard merupakan penggabungan informasi dan divisualisasikan ke dalam satu layar guna mencapai tujuan tertentu (Few, 2006).

Dashboard sendiri merupakan alat *monitoring* organisasi yang berisi tampilan informasi dari waktu ke waktu, dimana informasi ini berguna bagi pihak manajerial sebagai pihak pengambil keputusan untuk secara aktif dapat mengakses informasi kinerja bisnis tersebut lewat suatu *user interface* (Orts, 2005).

Dashboard merupakan mekanisme visualisasi informasi dalam sistem manajemen kinerja, yang menyajikan informasi kinerja operasional (Eckerson, 2005).

Dari beberapa penjelasan yang telah di paparkan di atas, dapat disimpulkan bahwa *dashboard* merupakan *tools* yang mampu memberikan visualisasi informasi ke dalam satu layar.

2.14.2 Tujuan Penggunaan *Dashboard*

Dashboard dapat dimanfaatkan sebagai *tools* untuk memonitor sistem secara kompleks dan *interdependent* (Malik, 2005). Untuk menentukan sebuah kebijakan dan strategi guna mendukung pencapaian tujuan organisasi, pihak manajerial membutuhkan berbagai informasi penting. Sedangkan Eckerson (2005) mengungkapkan bahwa *dashboard* dapat dimanfaatkan guna membantu dalam pencapaian tujuan strategis organisasi.

Penggunaan *dashboard* bertujuan untuk mengukur, memonitor, dan memprediksi suatu aktivitas baik yang telah berjalan, sedang berjalan, maupun aktivitas selanjutnya. Oleh karena itu, organisasi diharapkan dapat mengukur dan memperbaiki strategi yang ada dalam rangka memaksimalkan kinerjanya. *Dashboard* juga memiliki beberapa kelebihan diantaranya (Eckerson, 2005):

1. Mengkomunikasikan strategi

Sebagai sarana pihak eksekutif untuk mengkomunikasikan strategi yang telah dicanangkan pada seluruh pihak yang berperan.

2. Memonitor dan menyusun kembali pelaksanaan strategi

Pengidentifikasi masalah kritis dan penentuan strategi sebagai solusi dapat segera dilakukan oleh pihak eksekutif melalui *monitoring* terhadap pelaksanaan strategi.

3. Menyampaikan informasi ke semua pihak

Persepsi informasi dapat lebih mudah dipahami secara tepat oleh pengguna lewat penyajian informasi dalam bentuk bagan, grafik, simbol, dan warna yang interaktif.

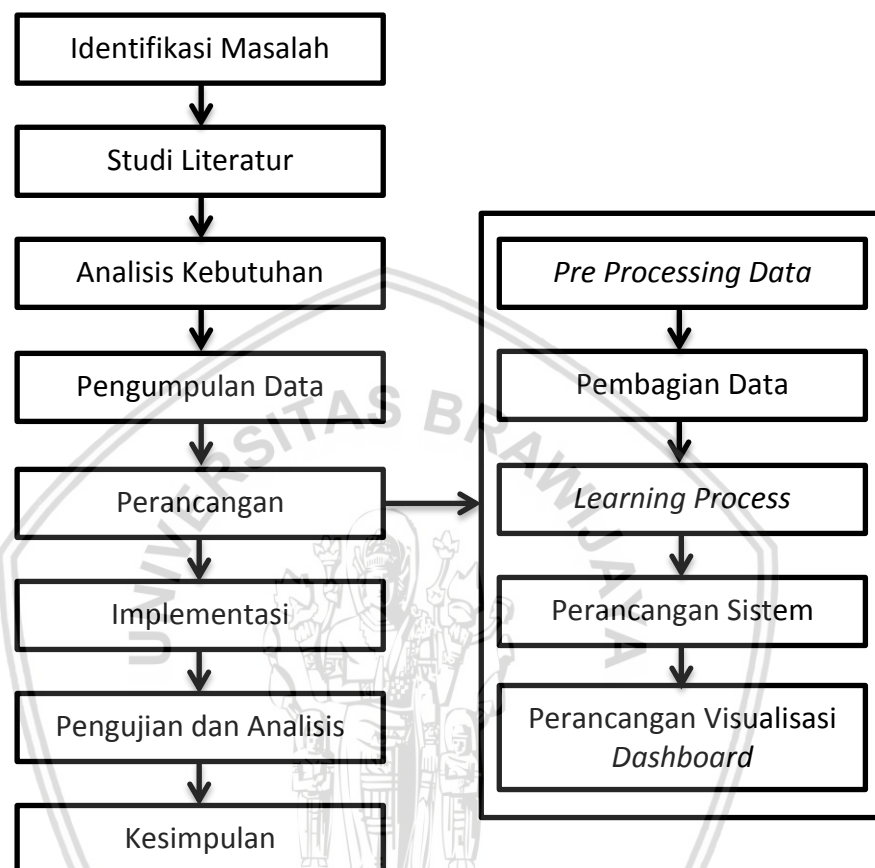
2.15 *Snake Case*

Snake case merupakan teknik penulisan kata majemuk atau frasa dimana elemen dipisahkan dengan satu karakter garis bawah (`_`) dan tidak ada spasi. Teknik penulisan ini dapat dilakukan dengan huruf besar atau kecil, tetapi harus konsisten dalam penggunaannya (Meissler, 2010). *Snake case* biasanya digunakan dalam kode komputer untuk nama variabel, nama fungsi, dan terkadang juga nama *file*.



BAB 3 METODOLOGI

Alur penelitian prediksi kredit macet melalui perilaku nasabah pada Koperasi Simpan Pinjam Mitra Raya Wates menggunakan metode klasifikasi C4.5 digambarkan dalam Gambar 3.1.



Gambar 3. 1 Alur Penelitian

3.1 Identifikasi Masalah

Pada tahap ini dilakukan pengidentifikasian masalah pada KSP Mitra Raya Wates. Masalah tersebut berkaitan dengan masalah belum adanya metode yang mampu memprediksi kredit macet yang mampu membantu dalam pengambilan keputusan bagi Pimpinan KSP Mitra Raya Wates. Proses pengidentifikasian masalah dilakukan dengan cara melakukan wawancara langsung dengan pihak manajerial dari KSP Mitra Raya Wates yang terlampir dalam Lampiran A.1.

3.2 Studi Literatur

Tahap selanjutnya adalah mencari berbagai literatur yang nantinya akan mendukung penelitian ini. Literatur-literatur yang digunakan bisa berasal dari buku, jurnal atau paper, atau penelitian yang telah dilakukan sebelumnya. Dalam penelitian ini dijelaskan mengenai teori-teori mengenai penjurusan, *knowledge*

discovery in database (KDD) dan *data mining*, klasifikasi, *decision tree*, metode evaluasi dan validasi algoritme *data mining*, *weka*, analisis kebutuhan, perancangan sistem, UML, implementasi sistem, pengujian sistem, dan konsep dasar *dashboard*.

3.3 Analisis Kebutuhan

Dilakukan pengidentifikasian untuk mencari apa yang dibutuhkan dan apa yang harus bisa dilakukan sistem, baik berupa kebutuhan fungsional maupun kebutuhan non-fungsional. Proses untuk mengetahui kebutuhan dari user dilakukan dengan cara wawancara langsung kepada pihak manajerila KSP Mitra Raya Wates yang terlampir dalam Lampiran A.2. Hasil luaran dari analisis berupa dokumen SRS (*System Requirement Specification*) untuk sistem yang akan dibangun.

3.4 Pengumpulan Data

Proses pengumpulan data dilakukan dengan cara ekstraksi data dari *database* dalam format *.xls*. Data yang didapatkan berjumlah 193 data yang didapatkan dari perpaduan dua *source data* yaitu *database* yang menampung data terkait pinjaman nasabah beserta dengan status pinjamannya dan juga data fisik yang tersimpan di gudang penyimpanan data yang terdiri dari data nasabah beserta dengan preferensinya. Hasil ekstraksi data ini akan digunakan untuk menentukan atribut yang diperlukan untuk melakukan prediksi kredit macet berdasarkan preferensi nasabah. Jenis data yang digunakan dalam melakukan prediksi kredit macet berdasarkan preferensi nasabah yaitu:

1. Data nasabah yang dilengkapi dengan data tanggungan keluarga, jumlah pendapatan, jumlah pengeluaran, jumlah pinjaman dana, agunan, dan status macet tidaknya nasabah tersebut.
2. Data yang digunakan adalah data pada bulan Oktober – Desember 2017 dan bulan Januari 2018.
3. Data pendukung lainnya.

3.5 Perancangan

Setelah data yang dibutuhkan untuk penelitian ini telah terkumpul, maka tahap selanjutnya adalah tahap perancangan. Tahap pertama yang dilakukan adalah melakukan *pre processing data*. Pada proses *pre processing data*, dilakukan proses *data selection* terlebih dahulu guna melakukan perubahan nama atribut dengan menggunakan *snake case style* agar nantinya setiap atribut yang digunakan mudah dikenali oleh *tool Weka*. Selain itu dilakukan proses pemilihan atribut yang akan digunakan untuk melakukan prediksi nasabah pada KSP Mitra Raya Wates. Jumlah atribut yang digunakan berjumlah tujuh atribut dari total 13 atribut pada *dataset* yang terkumpul. Pemilihan tujuh atribut didasarkan pada hasil wawancara dengan Pimpinan KSP Mitra Raya Wates, dimana tujuh atribut tersebut berpengaruh langsung terhadap macet tidaknya

pinjaman nasabah (terlampir dalam Lampiran A.1). Tujuh atribut yang digunakan tersebut juga berperan sebagai indikator yang digunakan untuk melakukan survei kepada nasabah yang mengajukan pinjaman.

Mengingat data yang didapatkan biasanya masih terdapat kurangnya nilai pada atribut, kurang memiliki atribut minat tertentu, hanya berisi data agregasi, terdapat *error* maupun *outlier*, dan juga terdapat perbedaan pada kode atau nama. Hal tersebut membuat *dataset* sangat tidak bersih karena terdapat *incomplete*, *noise*, dan juga *inconsistent data*. Maka diperlukan pembersihan data atau *data cleansing* sehingga dihasilkan data yang bersih, berkualitas, dan juga konsisten. Data yang bersih, berkualitas, dan juga konsisten akan menghasilkan proses klasifikasi yang berkualitas. Selain itu dilakukan juga proses transformasi pada *dataset* guna meningkatkan efisiensi dari model pembelajaran algoritme yang digunakan. Pada tahap ini data yang akan diolah akan ditransformasikan ke dalam bentuk kategori nilai yang sesuai. Kategori nilai didasarkan pada pendapat ahli yang dalam hal ini adalah Pimpinan KSP (terlampir dalam Lampiran C). Proses *data cleansing* dan *data transformation* dalam penelitian ini menggunakan *tool Microsoft Excel*.

Setelah dilakukan *pre processing data*, tahap selanjutnya adalah tahap membagi data menjadi dua bagian, yaitu kelompok data latih dan kelompok data uji. Data latih digunakan untuk melatih klasifikator, sedangkan data uji digunakan untuk memperkirakan *error rate* dari klasifikator yang telah dilatih. Dalam penelitian ini pembagian data dilakukan dengan persentase 80% untuk data latih dan 20% untuk data uji. Proses pembagian data pada penelitian ini memanfaatkan *tool Weka*.

Setelah tahap *pre processing data* dan pembagian data, perlu dibuat sebuah *rule* dalam bentuk pohon keputusan yang didapatkan dari atribut-atribut data latih yang digunakan sebagai parameter untuk memprediksi kredit macet berdasarkan preferensi nasabah. Untuk membuat sebuah pohon keputusan langkah pertama adalah menghitung akar pohon yang diambil dari atribut yang akan terpilih dengan cara menghitung nilai *gain* dari masing-masing atribut, nilai *gain* yang paling tinggi akan menjadi akar pertama. Sebelum menghitung nilai *gain* dari atribut, dilakukan perhitungan terlebih dahulu untuk nilai *entropy*. Proses tersebut diulang-ulang hingga semua *record* telah terpartisi. Setelah semua *record* telah terpartisi maka akan didapatkan sebuah *rule* dalam bentuk pohon keputusan yang nantinya bisa digunakan untuk mengklasifikasi data baru.

Selanjutnya, untuk mendapatkan *rule* berupa pohon keputusan yang maksimal, maka dilakukan proses *learning* dengan menggunakan *tool Weka*. Pada tahap ini, *data training* digunakan sebagai *input* agar *Weka* dapat memproses pembentukan *rule* dalam bentuk pohon keputusan berdasarkan metode C4.5. Tahap ini lebih bertujuan untuk memperoleh *rule* dalam bentuk pohon keputusan yang nantinya akan diimplementasikan sebagai sistem berbasis *web* yang dapat memprediksi kredit macet berdasarkan preferensi nasabah pada KSP Mitra Raya Wates.

Untuk membuat sebuah sistem yang dapat bekerja secara sempurna, maka diperlukan perancangan sistem terlebih dahulu. Dalam tahap perancangan ini akan mengonversikan dokumen SRS ke dalam bentuk bahasa perancangan UML dengan memanfaatkan *use case diagram*, *sequence diagram*, dan juga *class diagram*.

Tahap terakhir adalah desain antarmuka visualisasi *dashboard*. Tahap ini dilakukan agar antarmuka nantinya dapat sesuai dengan keinginan *client* dan tentu saja dapat memudahkan *client* dalam menggunakan sistem yang dihasilkan nantinya.

3.6 Implementasi

Pada tahap implementasi ini sistem dibangun sesuai perancangan yang telah dibuat dengan menerapkan studi literatur yang ada. Hasil dari tahap perancangan diimplementasikan dalam bentuk bahasa pemrograman PHP berbasis OOP dengan memanfaatkan fitur-fitur yang disediakan *Weka*. *Tools Weka* tersebut akan dikombinasikan dengan *framework codeigniter* untuk menghasilkan sebuah sistem. Selain itu dilakukan implementasi desain antarmuka visualisasi *dashboard* agar sistem dapat bekerja secara sempurna nantinya. Keluaran yang dihasilkan dari tahap implementasi ini adalah sistem rekomendasi yang dapat memberikan prediksi kredit macet pada pengajuan kredit nasabah.

3.7 Pengujian dan Analisis

Pada tahap ini dilakukan pengujian sistem yang telah dibangun. Pengujian sistem didasarkan dari pengujian *blackbox* atau berdasarkan pada fungsional sistem. Selain itu dilakukan juga pengevaluasian dan pemvalidasian terhadap metode C4.5 yang digunakan. Proses pengevaluasian dan pemvalidasian metode algoritme C4.5 dilakukan dengan cara menginputkan data uji pada *Weka* yang akan menghasilkan *confussion matrix* dan kurva ROC. *Confussion matrix* merepresentasikan tingkat akurasi dari proses klasifikasi yang telah dilakukan. Sedangkan Kurva ROC/AUC merupakan representasi perbandingan klasifikasi secara visual. Kurva ROC/AUC juga digunakan untuk menentukan apakah hasil klasifikasi dengan menggunakan C4.5 untuk memprediksi kredit macet berdasarkan preferensi nasabah pada KSP Mitra Raya Wates termasuk metode klasifikasi yang baik atau tidak.

Tahap selanjutnya adalah menganalisis hasil pengujian sistem serta pengevaluasian dan pemvalidasian metode C4.5. Analisis yang dilakukan berdasarkan pada hasil yang didapatkan dari pengujian *blackbox*, pengevaluasian dan pemvalidasian dengan *confussion matrix* dan kurva ROC. Pada hasil *confussion matrix* akan dilakukan pembahasan dari tingkat akurasi yang didapatkan serta analisis terkait data yang mengalami kesalahan klasifikasi berdasarkan dari *rule* yang telah terbentuk dari proses *learning* sebelumnya. Sedangkan dari kurva ROC akan dibahas mengenai nilai AUC yang didapatkan.

Luaran berupa sistem rekomendasi untuk memprediksi kredit macet berdasarkan preferensi nasabah akan diujikan langsung pada pihak KSP Mitra Raya Wates dengan menggunakan kuesioner *System Usability Scale* (SUS) guna menilai kualitas sistem berdasarkan penilaian dari KSP Mitra Raya Wates.

3.8 Penarikan Kesimpulan dan Saran

Tahap terakhir adalah penarikan kesimpulan berdasarkan rumusan masalah. Kemudian pemberian saran untuk penelitian di masa mendatang.



BAB 4 PERANCANGAN

Pada bab ini dijelaskan tentang analisis kebutuhan sistem, pengumpulan data, *pre processing* data, proses pembelajaran C4.5, perancangan sistem, dan perancangan visualisasi *dashboard*.

4.1 Analisis Kebutuhan Sistem

Proses analisis kebutuhan sistem dilakukan dengan cara wawancara secara langsung pada KSP Mitra Raya Wates. Hasil dari wawancara terlampir dalam Lampiran A.2 dan akan menghasilkan dokumen SRS yang terbagi menjadi kebutuhan fungsional dan kebutuhan non-fungsional.

4.1.1 Kebutuhan Fungsional

Kebutuhan fungsional mendeskripsikan apa saja yang harus dikerjakan sistem untuk memenuhi kebutuhan pengguna. Kebutuhan fungsional dinyatakan dalam *software requirement statement* seperti yang tertera pada Tabel 4.1.

Tabel 4. 1 Kebutuhan Fungsional

Kode SRS	Kode Aktivitas	Aktor	<i>Software Requirement Statement</i>	<i>Level</i>
F-SRPK-1	BP-SRPK-1	Pimpinan KSP dan Admin KSP	Sistem harus bisa menangani pemasukan <i>dataset</i> tanpa ada data yang terduplikasi pada <i>database</i>	Normal
F-SRPK-2	BP-SRPK-2	Pimpinan KSP dan Admin KSP	Sistem harus bisa menampilkan keseluruhan data pengajuan kredit nasabah beserta dengan preferensinya	Normal
F-SRPK-3	BP-SRPK-3	Pimpinan KSP	Sistem harus bisa menampilkan form untuk masukan data nasabah yang mengajukan pinjaman	Normal
F-SRPK-4	BP-SRPK-4	Pimpinan KSP	Sistem harus bisa menampilkan hasil prediksi terhadap pengajuan kredit nasabah	Normal
F-SRPK-5	BP-SRPK-5	Pimpinan KSP dan Admin KSP	Sistem harus bisa menampilkan jumlah pengajuan nasabah beserta dengan statistiknya, baik yang diterima maupun yang ditolak dalam bentuk persentase, <i>time-series</i> , maupun <i>trend</i> tiap bulan	Normal
F-SRPK-	BP-SRPK-	Admin	Sistem harus bisa menampilkan	Normal

Kode SRS	Kode Aktivitas	Aktor	Software Requirement Statement	Level
6	6	KSP	tingkat akurasi	
F-SRPK-7	BP-SRPK-7	Admin KSP	Sistem harus bisa menampilkan pohon keputusan	Normal

Pada Tabel 4.1 dapat diketahui terdapat tujuh kebutuhan fungsional yang dijelaskan berdasarkan *software requirement statement*, diantaranya adalah sistem harus bisa menangani pemasukan *dataset* tanpa ada data yang terduplikasi pada *database* dimana aktor yang bisa melakukan fungsi tersebut adalah Pimpinan KSP dan Admin KSP yang ditandai dengan kode SRS F-SRPK-1, sistem harus bisa menampilkan data pengajuan kredit nasabah beserta dengan preferensinya dimana aktor yang bisa melakukan fungsi tersebut adalah Pimpinan KSP dan Admin KSP yang ditandai dengan kode SRS F-SRPK-2, sistem harus bisa menampilkan form untuk masukan data nasabah yang mengajukan pinjaman dimana aktor yang bisa melakukan fungsi tersebut adalah Pimpinan KSP yang ditandai dengan kode SRS F-SRPK-3, sistem harus bisa menampilkan hasil prediksi terhadap pengajuan kredit nasabah dimana aktor yang bisa melakukan fungsi tersebut adalah Pimpinan KSP yang ditandai dengan kode SRS F-SRPK-4, sistem harus bisa menampilkan jumlah pengajuan nasabah beserta dengan statistiknya, baik yang diterima maupun yang ditolak dalam bentuk persentase, *time-series*, maupun *trend* tiap bulan dimana aktor yang bisa melakukan fungsi tersebut adalah Pimpinan KSP dan Admin KSP yang ditandai dengan kode SRS F-SRPK-5, sistem harus bisa menampilkan tingkat akurasi dimana aktor yang bisa melakukan fungsi tersebut adalah Admin KSP yang ditandai dengan kode SRS F-SRPK-6, dan sistem harus bisa menampilkan pohon keputusan dimana aktor yang bisa melakukan fungsi tersebut adalah Admin KSP yang ditandai dengan kode SRS F-SRPK-7.

4.1.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional mendeskripsikan perilaku yang harus dimiliki sistem seperti kinerja dan kemudahan pengguna yang dinyatakan dalam *software requirement statement* seperti yang tertera pada Tabel 4.2.

Tabel 4. 2 Kebutuhan Non-Fungsional

Kode SRS	Software Requirement Statement	Level
NF-SRPK-1	<i>Reliability</i> - Sistem dapat berfungsi dengan baik	Normal
NF-SRPK-2	<i>Usability</i> – Sistem dapat digunakan dengan mudah oleh pengguna	Normal
NF-SRPK-3	<i>Security</i> - Sistem dapat membedakan fitur berdasarkan jenis penggunaanya	Normal

Pada Tabel 4.2 dapat diketahui terdapat tiga kebutuhan non-fungsional yang dijelaskan berdasarkan *software requirement statement*, diantaranya adalah sistem dapat berfungsi dengan baik yang ditandai dengan kode SRS NF-SRPK-1, sistem dapat digunakan dengan mudah oleh pengguna yang ditandai dengan kode SRS NF-SRPK-2, dan sistem dapat membedakan fitur berdasarkan jenis penggunaannya yang ditandai dengan kode NF-SRPK-3.

4.1.3 Use Case Diagram

Use case dapat diidentifikasi berdasarkan kebutuhan fungsional dan non-fungsional. *Use Case* untuk sistem yang akan dibangun dapat dilihat pada Tabel 4.3.

Tabel 4. 3 Identifikasi *Use Case*

Kode <i>Use Case</i>	Kode SRS	<i>Use Case</i>
UC-SRPK-1	F-SRPK-1	Memasukkan <i>Dataset</i>
UC-SRPK-2	F-SRPK-2	Melihat Data Pengajuan Nasabah
UC-SRPK-3	F-SRPK-3	Memasukkan Data Pengaju Pinjaman
UC-SRPK-4	F-SRPK-4	Melihat Hasil Prediksi
UC-SRPK-5	F-SRPK-5	Melihat Jumlah Pengajuan
UC-SRPK-6	F-SRPK-5	Melihat Persentase Pengajuan
UC-SRPK-7	F-SRPK-5	Melihat <i>Time-Series</i> Pengajuan
UC-SRPK-8	F-SRPK-5	Melihat <i>Trend</i> Pengajuan
UC-SRPK-9	F-SRPK-6	Melihat Tingkat Akurasi
UC-SRPK-10	F-SRPK-7	Melihat Pohon Keputusan
UC-SRPK-11	NF-SRPK-3	Autentifikasi
UC-SRPK-12	NF-SRPK-3	Keluar

Pada Tabel 4.3 dapat diketahui terdapat 12 *use case*, diantaranya adalah memasukkan *dataset* yang ditandai dengan kode UC-SRPK-1 dan didapatkan dari kebutuhan fungsional dengan kode F-SRPK-1, melihat data pengajuan nasabah yang ditandai dengan kode UC-SRPK-2 dan didapatkan dari kebutuhan fungsional dengan kode F-SRPK-2, memasukkan data pengaju pinjaman yang ditandai dengan kode F-SRPK-3 dan didapatkan dari kebutuhan fungsional dengan kode F-SRPK-3, melihat hasil prediksi yang ditandai dengan kode UC-SRPK-4 dan didapatkan dari kebutuhan fungsional dengan kode F-SRPK-4, melihat jumlah pengajuan yang ditandai dengan kode UC-SRPK-5 dan didapatkan dari kebutuhan fungsional dengan kode F-SRPK-5, melihat persentase pengajuan yang ditandai dengan kode UC-SRPK-6 dan didapatkan dari kebutuhan fungsional dengan kode F-SRPK-5, melihat *time-series* pengajuan yang ditandai dengan kode UC-SRPK-7 dan didapatkan dari kebutuhan fungsional dengan kode F-SRPK-5, melihat *trend* pengajuan yang ditandai dengan kode UC-SRPK-8 dan didapatkan dari kebutuhan fungsional dengan kode F-SRPK-5, melihat tingkat akurasi yang ditandai dengan kode UC-SRPK-9 dan didapatkan dari kebutuhan fungsional dengan kode F-SRPK-6, melihat pohon keputusan yang ditandai dengan kode UC-SRPK-10 dan didapatkan dari kebutuhan fungsional dengan kode F-SRPK-7, autentifikasi yang ditandai dengan kode UC-SRPK-11 dan didapatkan dari kebutuhan non-fungsional dengan kode NF-SRPK-3, dan keluar yang ditandai dengan kode UC-SRPK-12 dan didapatkan dari kebutuhan non-fungsional dengan kode F-SRPK-3.

Dalam ilustrasi dari sudut pandang perlakuan aktor terhadap sistem, dapat digambarkan menggunakan *use case diagram* yang terlampir dalam Lampiran C.

4.1.4 Use Case Scenario

Dalam *use case scenario* akan menggambarkan alur jalan dari sisi actor dan sistem. *Use case scenario* didapatkan dengan menganalisis *use case diagram*.

4.1.4.1 Use Case Scenario Memasukkan Dataset (UC-SRPK-1)

Tabel 4. 4 Use Case Scenario Memasukkan Dataset

<i>Use Case Name</i>	Memasukkan <i>dataset</i>
<i>Brief Description</i>	Merupakan proses untuk menginputkan <i>dataset</i> ke dalam sistem, sehingga dapat disimpan dan diakses
<i>Actor</i>	Pimpinan KSP / Admin KSP
<i>Pre-Condition</i>	Pimpinan KSP / Admin KSP telah <i>login</i> ke sistem
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Pimpinan KSP / Admin KSP mengakses menu <i>input dataset</i> 2. Sistem menampilkan halaman <i>input dataset</i> 3. Pimpinan KSP / Admin KSP memilih <i>dataset</i> yang ingin diunggah 4. Pimpinan KSP / Admin KSP melakukan submit <i>dataset</i>

	5. Sistem memproses sekaligus menyimpan <i>dataset</i> 6. Sistem menampilkan <i>warning success</i> 7. Pimpinan KSP / Admin KSP melihat <i>warning success</i>
<i>Alternative Flows</i>	Alt1 Sistem gagal menyimpan <i>dataset</i> Jika pada langkah 4 sistem gagal menyimpan <i>dataset</i> , maka ketika proses <i>submit</i> , sistem akan menampilkan pesan <i>error</i> jika pengunggahan <i>dataset</i> gagal
<i>Post Condition</i>	1. Penambahan data nasabah, data transaksi, <i>data train</i> , dan <i>data test</i> pada <i>database</i> 2. Pengonversian data menjadi <i>file arff</i> dan model

Tabel 4.4 mendeskripsikan *use case scenario* memasukkan *dataset* yang terdiri dari *use case name*, *brief description*, *actor*, *pre-condition*, *basic flow*, *alternative flows*, dan juga *post condition*.

4.1.4.2 Use Case Scenario Melihat Data Pengaju Nasabah (UC-SRPK-2)

Tabel 4. 5 Use Case Scenario Melihat Data Pengaju Nasabah

<i>Use Case Name</i>	Melihat data pengajuan nasabah
<i>Brief Description</i>	Merupakan proses untuk melihat data pengajuan kredit nasabah beserta preferensinya yang telah tersimpan di <i>database</i>
<i>Actor</i>	Pimpinan KSP / Admin KSP
<i>Pre-Condition</i>	1. Pimpinan KSP / Admin KSP telah <i>login</i> ke sistem 2. Pimpinan KSP / Admin KSP telah memasukkan <i>dataset</i> ke sistem
<i>Basic Flow</i>	1. Pimpinan KSP / Admin KSP mengakses menu data nasabah 2. Sistem menampilkan halaman lihat data nasabah 3. Pimpinan KSP / Admin KSP melihat data pengajuan kredit nasabah beserta dengan preferensinya
<i>Alternative Flows</i>	-
<i>Post Condition</i>	-

Tabel 4.5 mendeskripsikan *use case scenario* melihat data nasabah yang terdiri dari *use case name*, *brief description*, *actor*, *pre-condition*, *basic flow*, *alternative flows*, dan juga *post condition*.

4.1.4.3 Use Case Scenario Memasukkan Data Pengaju Pinjaman (UC-SRPK-3)

Tabel 4. 6 Use Case Scenario Memasukkan Data Pengaju Pinjaman

Use Case Name	Memasukkan data pengaju pinjaman
Brief Description	Merupakan proses untuk memasukkan data nasabah yang mengajukan pinjaman dengan cara mengisi berbagai informasi terkait nasabah ke <i>form</i> yang telah tersedia
Actor	Pimpinan KSP
Pre-Condition	<ol style="list-style-type: none"> 1. Pimpinan KSP telah <i>login</i> ke sistem 2. Pimpinan KSP / Admin KSP telah memasukkan <i>dataset</i> ke sistem
Basic Flow	<ol style="list-style-type: none"> 1. Pimpinan KSP mengakses menu prediksi nasabah 2. Sistem menampilkan halaman <i>input data</i> pengaju pinjaman berupa formulir 3. Pimpinan KSP mengisi setiap <i>field</i> berdasarkan dari informasi nasabah pengaju pinjaman yang bersangkutan dengan lengkap 4. Pimpinan KSP melakukan <i>submit form</i> 5. Sistem memproses sekaligus menyimpan masukan dari Pimpinan KSP
Alternative Flows	<p>Alt1 Pimpinan KSP tidak mengisi formulir dengan lengkap</p> <p>Jika pada langkah 3 Pimpinan KSP tidak mengisi formulir dengan lengkap, maka ketika proses submit, sistem akan menampilkan warning untuk mengisi bagian dari formulir yang kosong</p>
Post Condition	<ol style="list-style-type: none"> 1. Penambahan <i>data test</i> dan data pengajuan pada <i>database</i> 2. Pengonversian data menjadi <i>file arrf</i>

Tabel 4.6 mendeskripsikan *use case scenario* memasukkan data pengaju pinjaman yang terdiri dari *use case name*, *brief description*, *actor*, *pre-condition*, *basic flow*, *alternative flows*, dan juga *post condition*.

4.1.4.4 Use Case Scenario Melihat Hasil Prediksi (UC-SRPK-4)

Tabel 4. 7 Use Case Scenario Melihat Hasil Prediksi

Use Case Name	Melihat hasil prediksi
Brief Description	Merupakan proses untuk melihat hasil prediksi dari nasabah yang ingin mengajukan pinjaman apakah

	pengajuan dari nasabah tersebut tergolong sebagai pengajuan kredit yang lancar atau macet
<i>Actor</i>	Pimpinan KSP
<i>Pre-Condition</i>	<ol style="list-style-type: none"> 1. Pimpinan KSP telah <i>login</i> ke sistem 2. Pimpinan KSP / Admin KSP telah memasukkan <i>dataset</i> ke sistem
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Pimpinan KSP melakukan <i>submit form</i> pada halaman prediksi nasabah 2. Sistem memproses masukan dari Pimpinan KSP 3. Sistem akan menampilkan hasil prediksi dari data yang dimasukkan oleh Pimpinan KSP 4. Pimpinan KSP melihat hasil prediksi
<i>Alternative Flows</i>	-
<i>Post Condition</i>	-

Tabel 4.7 mendeskripsikan *use case scenario* melihat hasil prediksi yang terdiri dari *use case name*, *brief description*, *actor*, *pre-condition*, *basic flow*, *alternative flows*, dan juga *post condition*.

4.1.4.5 Use Case Scenario Melihat Jumlah Pengajuan (UC-SRPK-5)

Tabel 4. 8 Use Case Scenario Melihat Jumlah Pengajuan

<i>Use Case Name</i>	Melihat jumlah pengajuan
<i>Brief Description</i>	Merupakan proses untuk melihat jumlah pengajuan, baik pengajuan yang diterima maupun yang ditolak
<i>Actor</i>	Pimpinan KSP / Admin KSP
<i>Pre-Condition</i>	Pimpinan KSP / Admin KSP telah login ke sistem
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Pimpinan KSP / Admin KSP mengakses menu <i>dashboard</i> 2. Sistem menampilkan halaman <i>dashboard</i> beserta dengan jumlah pengajuan yang telah dilakukan 3. Pimpinan KSP / Admin KSP melihat jumlah pengajuan yang telah dilakukan
<i>Alternative Flows</i>	-
<i>Post Condition</i>	-

Tabel 4.8 mendeskripsikan *use case scenario* melihat jumlah pengajuan yang terdiri dari *use case name*, *brief description*, *actor*, *pre-condition*, *basic flow*, *alternative flows*, dan juga *post condition*.

4.1.4.6 Use Case Scenario Melihat Persentase Pengajuan (UC-SRPK-6)

Tabel 4. 9 Use Case Scenario Melihat Persentase Pengajuan

<i>Use Case Name</i>	Melihat persentase pengajuan
<i>Brief Description</i>	Merupakan proses untuk melihat pengajuan, baik pengajuan yang diterima maupun yang ditolak dalam bentuk persentase yang tergambarkan dalam <i>pie chart</i>
<i>Actor</i>	Pimpinan KSP / Admin KSP
<i>Pre-Condition</i>	Pimpinan KSP / Admin KSP telah login ke sistem
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Pimpinan KSP / Admin KSP mengakses menu <i>dashboard</i> 2. Sistem menampilkan halaman <i>dashboard</i> beserta dengan persentase pengajuan yang telah dilakukan 3. Pimpinan KSP / Admin KSP melihat persentase pengajuan yang telah dilakukan dalam bentuk <i>pie chart</i>
<i>Alternative Flows</i>	-
<i>Post Condition</i>	-

Tabel 4.9 mendeskripsikan *use case scenario* melihat persentase pengajuan yang terdiri dari *use case name*, *brief description*, *actor*, *pre-condition*, *basic flow*, *alternative flows*, dan juga *post condition*.

4.1.4.7 Use Case Scenario Melihat Time-Series Pengajuan (UC-SRPK-7)

Tabel 4. 10 Use Case Scenario Melihat Time-Series Pengajuan

<i>Use Case Name</i>	Melihat <i>time-series</i> pengajuan
<i>Brief Description</i>	Merupakan proses untuk melihat pengajuan, baik pengajuan yang diterima maupun yang ditolak dalam bentuk <i>time-series</i> yang tergambarkan dalam <i>pie chart</i>
<i>Actor</i>	Pimpinan KSP / Admin KSP
<i>Pre-Condition</i>	Pimpinan KSP / Admin KSP telah login ke sistem
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Pimpinan KSP / Admin KSP mengakses menu

	<p><i>dashboard</i></p> <ol style="list-style-type: none"> 2. Sistem menampilkan halaman <i>dashboard</i> beserta dengan <i>time-series</i> pengajuan yang telah dilakukan 3. Pimpinan KSP / Admin KSP melihat <i>time-series</i> pengajuan yang telah dilakukan dalam bentuk <i>bar chart</i>
<i>Alternative Flows</i>	-
<i>Post Condition</i>	-

Tabel 4.10 mendeskripsikan *use case scenario* melihat *time-series* pengajuan yang terdiri dari *use case name*, *brief description*, *actor*, *pre-condition*, *basic flow*, *alternative flows*, dan juga *post condition*.

4.1.4.8 Use Case Scenario Melihat Trend Pengajuan (UC-SRPK-8)

Tabel 4. 11 Use Case Scenario Melihat Trend Pengajuan

<i>Use Case Name</i>	Melihat <i>trend</i> pengajuan
<i>Brief Description</i>	Merupakan proses untuk melihat pengajuan, baik pengajuan yang diterima maupun yang ditolak dalam bentuk <i>trend</i> tiap bulan yang tergambarkan dalam <i>line chart</i>
<i>Actor</i>	Pimpinan KSP / Admin KSP
<i>Pre-Condition</i>	Pimpinan KSP / Admin KSP telah login ke sistem
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Pimpinan KSP / Admin KSP mengakses menu <i>dashboard</i> 2. Sistem menampilkan halaman <i>dashboard</i> beserta dengan <i>trend</i> pengajuan yang telah dilakukan 3. Pimpinan KSP / Admin KSP melihat <i>trend</i> pengajuan yang telah dilakukan dalam bentuk <i>line chart</i>
<i>Alternative Flows</i>	-
<i>Post Condition</i>	-

Tabel 4.11 mendeskripsikan *use case scenario* melihat *trend* pengajuan yang terdiri dari *use case name*, *brief description*, *actor*, *pre-condition*, *basic flow*, *alternative flows*, dan juga *post condition*.

4.1.4.9 Use Case Scenario Melihat Tingkat Akurasi (UC-SRPK-9)

Tabel 4. 12 Use Case Scenario Melihat Tingkat Akurasi

<i>Use Case Name</i>	Melihat tingkat akurasi
<i>Brief Description</i>	Merupakan proses untuk melihat tingkat akurasi yang dihasilkan dari proses prediksi dari sistem yang dibuat
<i>Actor</i>	Admin KSP
<i>Pre-Condition</i>	<ol style="list-style-type: none"> 1. Admin KSP telah <i>login</i> ke sistem 2. Pimpinan KSP / Admin KSP telah memasukkan <i>dataset</i> ke sistem
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Admin KSP mengakses menu lihat akurasi 2. Sistem menampilkan halaman lihat akurasi beserta dengan tingkat akurasi sistem 3. Admin KSP melihat tingkat akurasi dari sistem yang dibuat
<i>Alternative Flows</i>	-
<i>Post Condition</i>	-

Tabel 4.12 mendeskripsikan *use case scenario* melihat tingkat akurasi yang terdiri dari *use case name*, *brief description*, *actor*, *pre-condition*, *basic flow*, *alternative flows*, dan juga *post condition*.

4.1.4.10 Use Case Scenario Melihat Pohon Keputusan (UC-SRPK-10)

Tabel 4. 13 Use Case Scenario Melihat Pohon Keputusan

<i>Use Case Name</i>	Melihat pohon keputusan
<i>Brief Description</i>	Merupakan proses untuk melihat pohon keputusan yang dihasilkan dari proses <i>learning</i> terhadap dataset yang telah diunggah
<i>Actor</i>	Admin KSP
<i>Pre-Condition</i>	<ol style="list-style-type: none"> 1. Admin KSP telah login ke sistem 2. Pimpinan KSP / Admin KSP telah memasukkan <i>dataset</i> ke sistem
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Admin KSP mengakses menu lihat <i>tree</i> 2. Sistem menampilkan halaman lihat <i>tree</i> beserta dengan <i>tree</i> yang dihasilkan dari proses <i>learning</i> 3. Pimpinan KSP melihat <i>tree</i> yang dihasilkan dari proses <i>learning</i>

<i>Alternative Flows</i>	-
<i>Post Condition</i>	-

Tabel 4.13 mendeskripsikan *use case scenario* melihat pohon keputusan yang terdiri dari *use case name*, *brief description*, *actor*, *pre-condition*, *basic flow*, *alternative flows*, dan juga *post condition*.

4.1.4.11 Use Case Scenario Autentifikasi (UC-SRPK-11)

Tabel 4. 14 Use Case Scenario Autentifikasi

<i>Use Case Name</i>	Autentifikasi
<i>Brief Description</i>	Merupakan proses untuk masuk ke dalam sistem
<i>Actor</i>	Pimpinan KSP / Admin KSP
<i>Pre-Condition</i>	Pimpinan KSP / Admin KSP telah terdaftar di <i>database</i>
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Pimpinan KSP / Admin KSP mengakses halaman login 2. Sistem menampilkan form login 3. Pimpinan KSP / Admin KSP mengisi form login dengan username dan password yang sesuai di database 4. Pimpinan KSP / Admin KSP mengakses menu masuk 5. Sistem menampilkan halaman <i>dashboard</i> 6. Pimpinan KSP / Admin KSP melihat halaman <i>dashboard</i>
<i>Alternative Flows</i>	<p>Alt1 Pimpinan KSP / Admin KSP mengisi form login dengan data yang tidak sesuai dengan data di database</p> <p>Jika pada langkah 3 Pimpinan KSP / Admin KSP mengisi form login dengan username dan password yang tidak sesuai dengan data di database, maka ketika mengakses menu masuk, sistem akan menampilkan pesan <i>error username dan password salah</i></p>
<i>Post Condition</i>	-

Tabel 4.14 mendeskripsikan *use case scenario* autentifikasi yang terdiri dari *use case name*, *brief description*, *actor*, *pre-condition*, *basic flow*, *alternative flows*, dan juga *post condition*.

4.1.4.12 Use Case Scenario Keluar (UC-SRPK-12)

Tabel 4. 15 Use Case Scenario Keluar

<i>Use Case Name</i>	Keluar
<i>Brief Description</i>	Merupakan proses untuk keluar dari sistem
<i>Actor</i>	Pimpinan KSP / Admin KSP
<i>Pre-Condition</i>	Pimpinan KSP / Admin KSP telah login ke sistem
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Pimpinan KSP / Admin KSP mengakses menu logout 2. Sistem menampilkan halaman login
<i>Alternative Flows</i>	-
<i>Post Condition</i>	-

Tabel 4.15 mendeskripsikan *use case scenario* keluar yang terdiri dari *use case name*, *brief description*, *actor*, *pre-condition*, *basic flow*, *alternative flows*, dan juga *post condition*.

4.2 Pengumpulan Data

Pengumpulan data dilakukan dengan cara menginputkan perpaduan antara data yang telah tersimpan pada *database* KSP Mitra Raya Wates (data pinjaman serta status pinjaman nasabah) dengan data fisik yang tersimpan di gudang penyimpanan dokumen KSP Mitra Raya Wates (data nasabah beserta dengan preferensinya) ke dalam *file excel*. Data yang terkumpul berjumlah 193 *records* yang merupakan data pada periode bulan Oktober 2017 hingga Januari 2018. *Dataset* yang telah terkumpul ditampilkan pada Lampiran D. Sedangkan atribut-atribut dari data yang terkumpul dapat dilihat pada Tabel 4.16.

Tabel 4. 16 Atribut Dataset

No	Nama Atribut	Keterangan
1	Nama Anggota	Nama Anggota
2	Jenis Kelamin	Jenis kelamin dari nasabah
3	Alamat	Alamat nasabah
4	Pekerjaan	Pekerjaan nasabah
5	Total Pinjaman	Total pinjaman nasabah
6	Waktu Angsuran	Waktu yang digunakan untuk mengangsur pinjaman nasabah
7	Tipe Agunan	Tipe agunan yang digunakan sebagai jaminan pinjaman nasabah

No	Nama Atribut	Keterangan
8	Jumlah Tanggungan	Jumlah tanggungan nasabah
9	Status Pernikahan	Status pernikahan nasabah
10	Bunga Pokok (6 Bulan)	Jumlah bunga pokok dari total pinjaman nasabah selama 6 bulan
11	Pendapatan Utama	Pendapatan utama nasabah
12	Total Biaya Hidup	Total biaya hidup dari nasabah
13	Status Pinjaman	Status nasabah lancar atau macet

Terdapat 13 atribut dari *dataset* yang didapatkan langsung dari KSP Mitra Raya Wates, diantaranya terdiri dari atribut Nama Anggota, Jenis Kelamin, Alamat, Pekerjaan, Total Pinjaman, Waktu Angsuran, Tipe Agunan, Jumlah Tanggungan, Status Pernikahan, Bunga Pokok (6 Bulan), Pendapatan Utama, Total Biaya Hidup, dan Status Pinjaman.

4.3 Pre Processing Data

Sebelum dilakukan proses pengklasifikasian, perlu dilakukan proses *pre processing data* guna mendapatkan data yang bersih dan berkualitas. Mengingat biasanya *dataset* yang telah terkumpul masih terdapat *incomplete*, *noise* dan juga *inconsistent data* serta beberapa penyesuaian terhadap atribut serta nilai di masing-masing atribut.

Ada empat tahap yang dilakukan pada proses *pre processing data* ini, diantaranya adalah *data selection*, *data cleansing*, *data transformation*, dan juga pembagian data.

4.3.1 Data Selection

Untuk memudahkan proses *learning* pada Weka, maka dilakukan perubahan nama atribut dari *dataset* yang telah terkumpul. *Style* yang digunakan untuk penamaan atribut baru menggunakan aturan *snake case*. Hasil dari perubahan atribut *dataset* ditampilkan pada Tabel 4.17.

Tabel 4. 17 Perubahan Atribut *Dataset*

No	Atribut Lama	Atribut Baru
1	Nama Anggota	NAMA_ANGGOTA
2	Jenis Kelamin	JENIS_KELAMIN
3	Alamat	ALAMAT
4	Pekerjaan	PEKERJAAN
5	Total Pinjaman	TOTAL_PINJAMAN

No	Atribut Lama	Atribut Baru
6	Waktu Angsuran	WAKTU_ANGSURAN
7	Tipe Agunan	TIPE_AGUNAN
8	Jumlah Tanggungan	JUMLAH_TANGGUNGAN
9	Status Pernikahan	STATUS_PERNIKAHAN
10	Bunga Pokok (6 Bulan)	BUNGA_POKOK(6_BULAN)
11	Pendapatan Utama	PENDAPATAN_UTAMA
12	Total Biaya Hidup	TOTAL_BIAYA_HIDUP
13	Status Pinjaman	STATUS_PINJAMAN

Pada Tabel 4.17 dijabarkan atribut-atribut yang akan digunakan dalam penelitian ini untuk memprediksi kredit macet pada KSP Mitra Raya Wates, yaitu JENIS_KELAMIN, TOTAL_PINJAMAN, JUMLAH_TANGGUNGAN, STATUS_PERNIKAHAN, PENDAPATAN_UTAMA, TOTAL_BIAYA_HIDUP yang nantinya akan digunakan sebagai atribut *predictor*, dan STATUS_PINJAMAN sebagai atribut *class*. Pemilihan atribut yang digunakan didasarkan pada hasil wawancara yang telah dilakukan langsung kepada Pimpinan KSP Mitra Raya Wates (terlampir dalam Lampiran C), dimana tujuh atribut tersebut berpengaruh langsung terhadap macet tidaknya pinjaman nasabah. Selain itu tujuh atribut tersebut juga digunakan sebagai indikator untuk melakukan survei kepada nasabah yang mengajukan pinjaman. Potongan data hasil pemilihan atribut ditampilkan pada Tabel 4.18.

Tabel 4. 18 Potongan Data Hasil Pemilihan Atribut

JENIS_KELAMIN	TOTAL_PINJAMAN	JUMLAH_TANGGUNGAN	STATUS_PERNIKAHAN	PENDAPATAN_UTAMA	TOTAL_BIAYA_HIDUP	STATUS_PINJAMAN
LAKI-LAKI	3500000	4	MENIKAH	1700000	1000000	MACET
PEREMPUAN	2000000	3	MENIKAH	2000000	1500000	LANCAR
LAKI-LAKI	4000000	0	BELUM MENIKAH	3000000	2000000	LANCAR
PEREMPUAN	2500000	2	MENIKAH	1500000	1000000	LANCAR
LAKI-LAKI	7000000	4	MENIKAH	2500000	1500000	MACET
LAKI-LAKI	2500000	3	MENIKAH	2500000	1500000	LANCAR

JENIS_KELAMIN	TOTAL_PINJAMAN	JUMLAH_TANGGUNGAN	STATUS_PERKAWINAN	PENDAPATAN_UTAMA	TOTAL_BIAYA_HIDUP	STATUS_PINJAMAN
LAKI-LAKI	5500000	4	MENIKAH	1500000	1000000	LANCAR
PEREMPUAN	7500000	2	MENIKAH	2000000	1000000	MACET
LAKI-LAKI	3500000	2	MENIKAH	1500000	1000000	LANCAR
PEREMPUAN	4300000	5	MENIKAH	1700000	1000000	MACET

4.3.2 Data Cleansing

Dataset yang telah terkumpul langsung dari KSP Mitra Raya Wates masih bersifat *incomplete*, mengingat terdapat 8 baris data yang masing-masing terdiri dari tiga sel pada atribut JUMLAH_TANGGUNGAN dan lima sel pada atribut PENDAPATAN_UTAMA dan TOTAL_BIAYA_HIDUP bernilai kosong (dilampirkan dalam Lampiran D). Hal tersebut mengharuskan dilakukannya *data cleansing* guna menghasilkan proses klasifikasi yang berkualitas dengan cara menghapus *missing values* pada delapan baris data tersebut. Proses penghapusan baris dilakukan secara manual memanfaatkan *tool Microsoft Excel*. Dari proses *data cleansing* yang telah dilakukan, data yang bersih dan dapat digunakan untuk proses klasifikasi berjumlah 185 *records*.

4.3.3 Data Transformation

Setelah dilakukan proses *data cleansing*, *pre processing data* yang dilakukan selanjutnya adalah melakukan transformasi data. Tujuan dilakukannya transformasi data pada penelitian ini adalah guna meningkatkan efisiensi dari model pembelajaran algoritme yang digunakan. Proses transformasi data dilakukan dengan cara merubah nilai pada atribut JENIS_KELAMIN, TOTAL_PINJAMAN, JUMLAH_TANGGUNGAN, STATUS_PERKAWINAN, PENDAPATAN_UTAMA, dan TOTAL_BIAYA_HIDUP menjadi bentuk nilai yang terkategori sesuai dengan atribut masing-masing berdasarkan pembagian kategori dari Pimpinan KSP Mitra Raya Wates (terlampir pada Lampiran D) dengan memanfaatkan *Microsoft Excel*. Hasil dari transformasi data yang dilakukan direpresentasikan pada Tabel 4.19.

Tabel 4. 19 Hasil Data Transformation

No	Atribut	Keterangan	Kategori Nilai
1	JENIS_KELAMIN	LAKI-LAKI	L
		PEREMPUAN	P
2	TOTAL_PINJAMAN	0 - 2500000	SANGAT_RENDAH
		2500001 - 4500000	RENDAH

No	Atribut	Keterangan	Kategori Nilai
		4500001 - 6000000	SEDANG
		6000001 - 8000000	TINGGI
		> 8000000	SANGAT_TINGGI
3	JUMLAH_TANGGUNGAN	0 – 2	SEDIKIT
		3 – 4	SEDANG
		> 4	BANYAK
4	STATUS_PERNIKAHAN	MENIKAH	MENIKAH
		BELUM MENIKAH	BELUM_MENIKAH
		CERAI	CERAI
5	PENDAPATAN_UTAMA	0 - 1500000	RENDAH
		1500001 - 3000000	SEDANG
		> 3000000	TINGGI
6	TOTAL_BIAYA_HIDUP	0 - 1500000	RENDAH
		1500001 - 3000000	SEDANG
		> 3000000	TINGGI

Pada Tabel 4.19, dapat diketahui bahwa untuk atribut JENIS_KELAMIN yang terdiri dari nilai LAKI-LAKI dirubah menjadi L dan nilai PEREMPUAN dirubah menjadi P. Atribut TOTAL_PINJAMAN dengan nilai pinjaman berkisar antara 0 sampai dengan 2500000 dirubah menjadi pinjaman dengan kategori SANGAT_RENDAH, nilai pinjaman berkisar antara 2500001 sampai dengan Rp 4500000 dirubah menjadi pinjaman dengan kategori RENDAH, nilai pinjaman berkisar antara 4500001 sampai dengan 6000000 menjadi pinjaman dengan kategori SEDANG, nilai pinjaman berkisar 6000001 sampai dengan 8000000 dirubah menjadi pinjaman dengan kategori TINGGI, dan nilai pinjaman di atas 8000000 dirubah menjadi pinjaman dengan kategori SANGAT_TINGGI. Pada atribut JUMLAH_TANGGUNGAN yang terdiri dari nilai 0 – 2 dirubah menjadi SEDIKIT, 3 – 4 dirubah menjadi SEDANG, dan nilai di atas 4 dirubah menjadi BANYAK. Atribut STATUS_PERNIKAHAN yang bernilai MENIKAH tidak mengalami perubahan, BELUM MENIKAH menjadi BELUM_MENIKAH, dan CERAI tidak mengalami perubahan. Dan pada atribut PENDAPATAN_UTAMA dan TOTAL_BIAYA_HIDUP dengan nilai berkisar antara 0 sampai 1500000 dirubah menjadi RENDAH, 1500001 sampai 3000000 dirubah menjadi SEDANG, dan nilai di atas 3000000 dirubah menjadi TINGGI.

Potongan data hasil dari proses transformasi data ditampilkan pada Tabel 4.20, sedangkan data hasil dari *pre processing data* secara keseluruhan ditampilkan dalam Lampiran E.

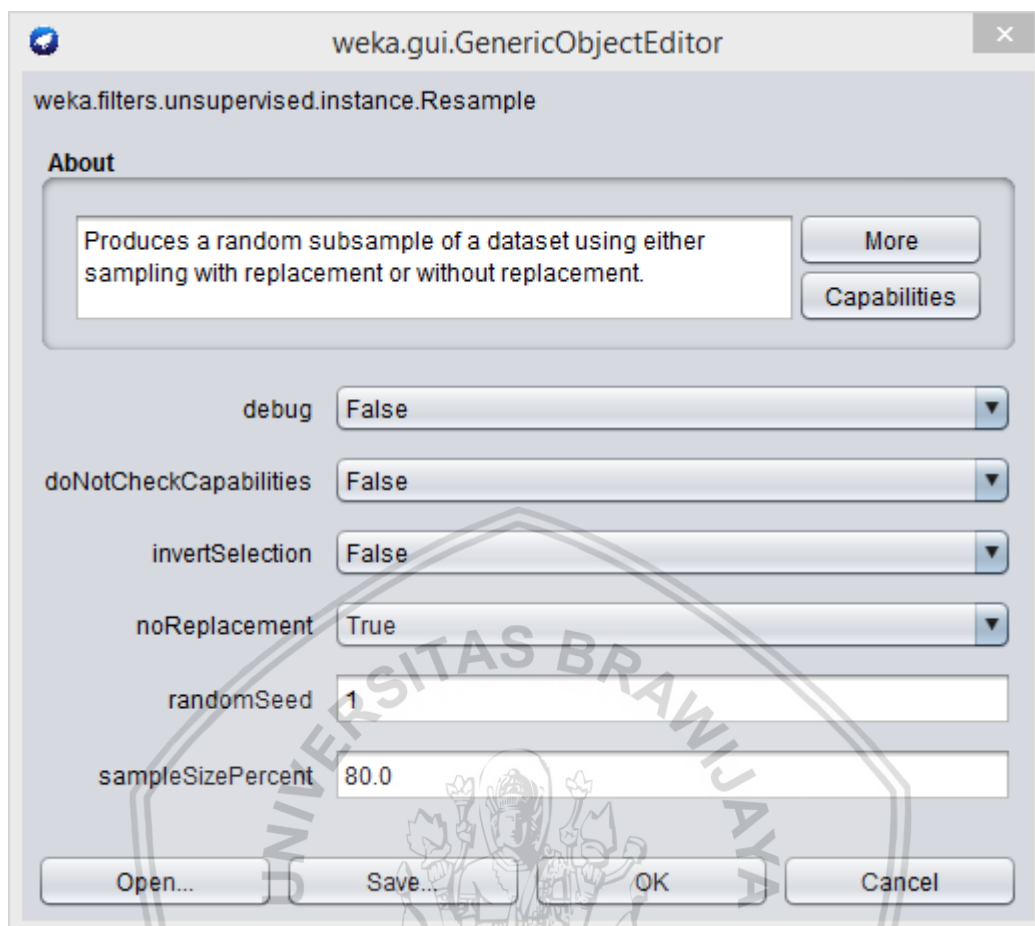
Tabel 4. 20 Potongan Data Hasil Transformation Data

JENIS_KELAMIN	TOTAL_PINJAMAN	JUMLAH_TANGGUNGAN	STATUS_PERNIKAHAN	PENDAPATAN_UTAMA	TOTAL_BIAYA_HIDUP	STATUS_PINJAMAN
L	RENDAH	SEDANG	MENIKAH	SEDANG	RENDAH	MACET

JENIS_K ELAMI N	TOTAL_P INJAMA N	JUMLAH_TA NGGUNGA N	STATUS_P ERNIKAHA N	PENDAPAT AN_UTAM A	TOTAL_BI AYA_HIDU P	STATUS_ PINJAMA N
P	SANGAT _RENDA H	SEDANG	MENIKAH	SEDANG	RENDAH	LANCAR
L	RENDAH	SEDIKIT	BELUM MENIKAH	SEDANG	SEDANG	LANCAR
P	SANGAT _RENDA H	SEDIKIT	MENIKAH	RENDAH	RENDAH	MACET
L	TINGGI	SEDANG	MENIKAH	SEDANG	RENDAH	MACET
L	SANGAT _RENDA H	SEDANG	MENIKAH	SEDANG	RENDAH	LANCAR
L	SEDANG	SEDANG	MENIKAH	RENDAH	RENDAH	MACET
P	TINGGI	SEDIKIT	MENIKAH	SEDANG	RENDAH	MACET
L	RENDAH	SEDIKIT	MENIKAH	RENDAH	RENDAH	LANCAR
P	RENDAH	BANYAK	MENIKAH	SEDANG	RENDAH	MACET

4.3.4 Pembagian Data

Setelah proses *pre processing data* telah selesai dilakukan, proses selanjutnya adalah proses pembagian data. Proses tersebut dilakukan dengan cara mengubah *dataset* yang sebelumnya berekstensi *excel* kemudian dikonversikan menjadi *file* berekstensi *arff* agar dapat dibaca oleh *Weka*. Proses pembagian data dilakukan dengan memanfaatkan fitur yang ada di *Weka* yaitu fitur *resample* dengan pilihan *randomSeed* adalah 1. Sehingga, pada saat dilakukan pembagian, dilakukan juga proses pengacakan terhadap data. Data yang telah mengalami proses *pre processing data* dibagi dalam persentase 80% : 20%. Dimana 80% untuk data latih dan 20% untuk data uji. Sehingga data yang dihasilkan untuk data latih berjumlah 148 *records* yang ditampilkan dalam Lampiran G untuk data latih. Sedangkan sisanya berjumlah 37 *records* sebagai data uji yang ditampilkan dalam Lampiran H untuk data uji. Proses pembagian data menggunakan *tool Weka* direpresentasikan dalam Gambar 4.1.



Gambar 4. 1 Proses Pembagian Data Menggunakan Fitur *Resample* pada *Weka*

4.4 Learning Process

Tahap ini menjelaskan proses pembelajaran guna menghasilkan *rule* dalam bentuk pohon keputusan berdasarkan algoritme C4.5. Ada dua tahap yang dilakukan pada proses *learning* ini, yaitu *learning process* secara manual yang menjelaskan proses untuk menghasilkan *rule* dalam bentuk pohon keputusan dan juga *learning process* menggunakan *Weka* guna menghasilkan *rule* berupa pohon keputusan yang nantinya dapat digunakan untuk memprediksi kredit macet berdasarkan preferensi nasabah pada KSP Mitra Raya Wates.

4.4.1 Learning Process Secara Manual

Proses pembelajaran secara manual ini lebih menjelaskan bagaimana sebuah *rule* dalam bentuk pohon keputusan menggunakan algoritme C4.5 dapat terbentuk. Data yang digunakan pada proses ini berjumlah 37 *records* yang didapatkan dari pemangkasan *dataset* secara keseluruhan. 37 data yang digunakan dapat dilihat pada Tabel 4.21. Berikut langkah-langkah dalam perhitungan algoritme C4.5:

Tabel 4. 21 Data untuk Perhitungan *Learning Process* Secara Manual

JENIS_K ELAMI N	TOTAL_PI NJAMAN	JUMLAH_T ANGGUN AN	STATUS_P ERNIKAHA N	PENDAPAT AN_UTAM A	TOTAL_BI AYA_HIDU P	STATUS_ PINJAMA N
L	SEDANG	SEDANG	MENIKAH	TINGGI	SEDANG	MACET
L	SANGAT_ RENDAH	SEDANG	MENIKAH	SEDANG	RENDAH	LANCAR
L	SANGAT_ TINGGI	SEDIKIT	BELUM_M ENIKAH	SEDANG	RENDAH	LANCAR
L	SANGAT_ RENDAH	SEDANG	MENIKAH	SEDANG	RENDAH	LANCAR
P	SEDANG	SEDANG	MENIKAH	SEDANG	SEDANG	MACET
L	SANGAT_ RENDAH	BANYAK	MENIKAH	SEDANG	SEDANG	MACET
P	SEDANG	SEDANG	MENIKAH	SEDANG	SEDANG	MACET
L	SANGAT_ RENDAH	SEDANG	MENIKAH	SEDANG	RENDAH	LANCAR
L	RENDAH	SEDANG	MENIKAH	SEDANG	RENDAH	MACET
P	RENDAH	SEDIKIT	MENIKAH	SEDANG	RENDAH	LANCAR
L	RENDAH	SEDIKIT	BELUM_M ENIKAH	SEDANG	RENDAH	LANCAR
P	RENDAH	SEDIKIT	MENIKAH	SEDANG	RENDAH	LANCAR
L	SANGAT_ TINGGI	SEDANG	MENIKAH	TINGGI	SEDANG	MACET
P	SEDANG	SEDIKIT	MENIKAH	SEDANG	RENDAH	MACET
L	RENDAH	SEDIKIT	BELUM_M ENIKAH	SEDANG	SEDANG	LANCAR
P	RENDAH	SEDIKIT	MENIKAH	SEDANG	RENDAH	LANCAR
P	RENDAH	SEDIKIT	MENIKAH	SEDANG	RENDAH	LANCAR
P	RENDAH	SEDANG	MENIKAH	SEDANG	SEDANG	MACET
P	RENDAH	BANYAK	MENIKAH	RENDAH	RENDAH	MACET
P	SANGAT_ RENDAH	SEDIKIT	MENIKAH	SEDANG	RENDAH	LANCAR
P	RENDAH	SEDIKIT	MENIKAH	SEDANG	RENDAH	LANCAR
L	RENDAH	SEDIKIT	MENIKAH	SEDANG	RENDAH	LANCAR
P	SEDANG	SEDANG	MENIKAH	SEDANG	RENDAH	MACET
P	RENDAH	SEDANG	MENIKAH	SEDANG	SEDANG	MACET
L	RENDAH	SEDIKIT	BELUM_M ENIKAH	SEDANG	RENDAH	LANCAR
L	SEDANG	SEDIKIT	CERAI	SEDANG	SEDANG	LANCAR
L	SANGAT_ RENDAH	SEDIKIT	MENIKAH	SEDANG	RENDAH	LANCAR
L	RENDAH	SEDANG	MENIKAH	SEDANG	RENDAH	MACET
P	RENDAH	SEDIKIT	MENIKAH	SEDANG	SEDANG	MACET
P	SANGAT_ RENDAH	SEDANG	CERAI	SEDANG	RENDAH	LANCAR

JENIS_KELAMIN	TOTAL_PINJAMAN	JUMLAH_TANGGUNGAN	STATUS_PERNIKAHAN	PENDAPATAN_UTAMA	TOTAL_BIAYA_HIDUP	STATUS_PINJAMAN
L	SEDANG	SEDANG	MENIKAH	SEDANG	RENDAH	LANCAR
L	SEDANG	SEDIKIT	BELUM_MENIKAH	SEDANG	SEDANG	LANCAR
L	SEDANG	SEDANG	MENIKAH	TINGGI	SEDANG	MACET
L	SEDANG	SEDANG	MENIKAH	RENDAH	RENDAH	MACET
L	SANGAT_RENDAH	SEDIKIT	MENIKAH	SEDANG	RENDAH	LANCAR
L	SANGAT_RENDAH	SEDIKIT	BELUM_MENIKAH	SEDANG	SEDANG	LANCAR
P	SEDANG	SEDIKIT	MENIKAH	SEDANG	RENDAH	LANCAR

Langkah 1: Menghitung nilai *entropy* dan *gain*

- Hitung *entropy* dari total kasus, yang mengacu pada Persamaan 2.2:

$$Entropy(total) = \left(-\frac{22}{37} * \log_2 \left(\frac{22}{37} \right) \right) + \left(-\frac{15}{37} * \log_2 \left(\frac{15}{37} \right) \right) = 0.974024864$$

- Kemudian hitung untuk nilai *entropy* pada setiap atribut, mengacu pada Persamaan 2.2:

$$Entropy(JENIS_KELAMIN, L) = \left(-\frac{14}{21} * \log_2 \left(\frac{14}{21} \right) \right) + \left(-\frac{7}{21} * \log_2 \left(\frac{7}{21} \right) \right) = 0.918296$$

$$Entropy(JENIS_KELAMIN, P) = \left(-\frac{8}{16} * \log_2 \left(\frac{8}{16} \right) \right) + \left(-\frac{8}{16} * \log_2 \left(\frac{8}{16} \right) \right) = 1$$

$$Entropy(TOTAL_PINJAMAN, SANGAT_RENDAH) = \left(-\frac{8}{9} * \log_2 \left(\frac{8}{9} \right) \right) + \left(-\frac{1}{9} * \log_2 \left(\frac{1}{9} \right) \right) = 0.503258$$

$$Entropy(TOTAL_PINJAMAN, RENDAH) = \left(-\frac{9}{15} * \log_2 \left(\frac{9}{15} \right) \right) + \left(-\frac{6}{15} * \log_2 \left(\frac{6}{15} \right) \right) = 0.970950594$$

$$Entropy(TOTAL_PINJAMAN, SEDANG) = \left(-\frac{4}{11} * \log_2 \left(\frac{4}{11} \right) \right) + \left(-\frac{7}{11} * \log_2 \left(\frac{7}{11} \right) \right) = 0.945660305$$

$$Entropy(TOTAL_PINJAMAN, TINGGI) = \left(-\frac{0}{0} * \log_2 \left(\frac{0}{0} \right) \right) + \left(-\frac{0}{0} * \log_2 \left(\frac{0}{0} \right) \right) = 0$$

$$\text{Entropy}(\text{TOTAL_PINJAMAN}, \text{SANGAT_TINGGI}) = \left(-\frac{1}{2} * \log_2\left(\frac{1}{2}\right)\right) + \left(-\frac{1}{2} * \log_2\left(\frac{1}{2}\right)\right) = 1$$

$$\text{Entropy}(\text{JUMLAH_TANGGUNGAN}, \text{SEDIKIT}) = \left(-\frac{17}{19} * \log_2\left(\frac{17}{19}\right)\right) + \left(-\frac{2}{19} * \log_2\left(\frac{2}{19}\right)\right) = 0.485460761$$

$$\text{Entropy}(\text{JUMLAH_TANGGUNGAN}, \text{SEDANG}) = \left(-\frac{5}{16} * \log_2\left(\frac{5}{16}\right)\right) + \left(-\frac{11}{16} * \log_2\left(\frac{11}{16}\right)\right) = 0.896038233$$

$$\text{Entropy}(\text{JUMLAH_TANGGUNGAN}, \text{BANYAK}) = \left(-\frac{0}{2} * \log_2\left(\frac{0}{2}\right)\right) + \left(-\frac{2}{2} * \log_2\left(\frac{2}{2}\right)\right) = 0$$

$$\text{Entropy}(\text{STATUS_PERNIKAHAN}, \text{MENIKAH}) = \left(-\frac{14}{29} * \log_2\left(\frac{14}{29}\right)\right) + \left(-\frac{15}{29} * \log_2\left(\frac{15}{29}\right)\right) = 0.999142104$$

$$\text{Entropy}(\text{STATUS_PERNIKAHAN}, \text{BELUM_MENIKAH}) = \left(-\frac{6}{6} * \log_2\left(\frac{6}{6}\right)\right) + \left(-\frac{0}{6} * \log_2\left(\frac{0}{6}\right)\right) = 0$$

$$\text{Entropy}(\text{STATUS_PERNIKAHAN}, \text{CERAI}) = \left(-\frac{2}{2} * \log_2\left(\frac{2}{2}\right)\right) + \left(-\frac{0}{2} * \log_2\left(\frac{0}{2}\right)\right) = 0$$

$$\text{Entropy}(\text{PENDAPATAN_UTAMA}, \text{RENDAH}) = \left(-\frac{0}{2} * \log_2\left(\frac{0}{2}\right)\right) + \left(-\frac{2}{2} * \log_2\left(\frac{2}{2}\right)\right) = 0$$

$$\text{Entropy}(\text{PENDAPATAN_UTAMA}, \text{SEDANG}) = \left(-\frac{22}{32} * \log_2\left(\frac{22}{32}\right)\right) + \left(-\frac{10}{32} * \log_2\left(\frac{10}{32}\right)\right) = 0.896038233$$

$$\text{Entropy}(\text{PENDAPATAN_UTAMA}, \text{TINGGI}) = \left(-\frac{0}{3} * \log_2\left(\frac{0}{3}\right)\right) + \left(-\frac{3}{3} * \log_2\left(\frac{3}{3}\right)\right) = 0$$

$$\text{Entropy}(\text{TOTAL_BIAYA_HIDUP}, \text{RENDAH}) = \left(-\frac{18}{24} * \log_2\left(\frac{18}{24}\right)\right) + \left(-\frac{6}{24} * \log_2\left(\frac{6}{24}\right)\right) = 0.811278124$$

$$\begin{aligned} \text{Entropy}(\text{TOTAL_BIAYA_HIDUP}, \text{SEDANG}) &= \left(-\frac{4}{13} * \log_2 \left(\frac{4}{13} \right) \right) + \\ &\left(-\frac{9}{13} * \log_2 \left(\frac{9}{13} \right) \right) = 0.89049164 \end{aligned}$$

$$\begin{aligned} \text{Entropy}(\text{TOTAL_BIAYA_HIDUP}, \text{TINGGI}) &= \left(-\frac{0}{0} * \log_2 \left(\frac{0}{0} \right) \right) + \\ &\left(-\frac{0}{0} * \log_2 \left(\frac{0}{0} \right) \right) = 0 \end{aligned}$$

- Setelah mencari nilai *entropy* pada setiap atribut, kemudian dilakukan **perhitungan nilai gain dari setiap atribut**, perhitungan nilai *gain* mengacu pada Persamaan 2.1:

$$\begin{aligned} \text{Gain}(\text{JENIS_KELAMIN}) &= 0.974024864 - \left(\left(\frac{21}{37} * 0.918296 \right) + \left(\frac{16}{37} * 1 \right) \right) \\ &= 0.020397499 \end{aligned}$$

$$\begin{aligned} \text{Gain}(\text{TOTAL_PINJAMAN}) &= 0.974024864 - \left(\left(\frac{9}{37} * 0.503258335 \right) + \right. \\ &\left. \left(\frac{15}{37} * 0.970950594 \right) + \left(\frac{11}{37} * 0.945660305 \right) + \left(\frac{0}{37} * 0 \right) + \left(\frac{2}{37} * 1 \right) \right) = \\ &0.122785749 \end{aligned}$$

$$\begin{aligned} \text{Gain}(\text{JUMLAH_TANGGUNGAN}) &= \\ 0.974024864 - \left(\left(\frac{19}{37} * 0.485460761 \right) + \left(\frac{16}{37} * 0.896038233 \right) + \right. \\ &\left. \left(\frac{2}{37} * 0 \right) \right) = 0.337258211 \end{aligned}$$

$$\begin{aligned} \text{Gain}(\text{STATUS_PERNIKAHAN}) &= 0.974024864 - \left(\left(\frac{29}{37} * 0.999142104 \right) + \right. \\ &\left. \left(\frac{6}{37} * 0 \right) + \left(\frac{2}{37} * 0 \right) \right) = 0.190913486 \end{aligned}$$

$$\begin{aligned} \text{Gain}(\text{PENDAPATAN_UTAMA}) &= 0.974024864 - \left(\left(\frac{2}{37} * 0 \right) + \right. \\ &\left. \left(\frac{32}{37} * 0.896038233 \right) + \left(\frac{3}{37} * 0 \right) \right) = 0.19907288 \end{aligned}$$

$$\begin{aligned} \text{Gain}(\text{TOTAL_BIAYA_HIDUP}) &= 0.974024864 - \left(\left(\frac{24}{37} * 0.811278124 \right) + \right. \\ &\left. \left(\frac{13}{37} * 0.89049164 \right) + \left(\frac{0}{37} * 0 \right) \right) = 0.134914964 \end{aligned}$$

- Hasil perhitungan *entropy* total, *entropy* tiap atribut, dan gain ditampilkan pada Tabel 4.22 dan 4.23. Secara lebih lengkap, perhitungan *entropy* tiap atribut ditampilkan dalam Lampiran H.1.

Tabel 4. 22 Perhitungan Nilai Entropy Total

Total Kasus	Sum(Lancar)	Sum(Macet)	Entropi Total
37	22	15	0.974024864

Tabel 4. 23 Perhitungan Nilai *Entropy* dan *Gain* Tiap Atribut untuk Penentuan *Node* Akar

Node	Atribut	Nilai	Entropi	Gain
1	JENIS_KELAMINAN			0.0203975
		L	0.918296	
		P	1	
	TOTAL_PINJAMAN			0.12278575
		SANGAT_RENDAH	0.503258	
		RENDAH	0.970951	
		SEDANG	0.94566	
		TINGGI	0	
		SANGAT_TINGGI	1	
	JUMLAH_TANGGUNGAN			0.33725821
		SEDIKIT	0.485461	
		SEDANG	0.896038	
		BANYAK	0	
	STATUS_PERNIKAHAN			0.19091349
		MENIKAH	0.999142	
		BELUM_MENIKAH	0	
		CERAI	0	
	PENDAPATAN_UTAMA			0.19907288
		RENDAH	0	
		SEDANG	0.896038	
		TINGGI	0	
	TOTAL_BIAYA_HIDUP			0.13491496
		RENDAH	0.811278	
		SEDANG	0.890492	
		TINGGI	0	

Langkah 2: Menentukan *node* akar

Berdasarkan perhitungan nilai *entropy* dan *gain* pada Tabel 4.23, dapat disimpulkan bahwa atribut JUMLAH_TANGGUNGAN merupakan *node* akar.

Penentuan *node* akar didasarkan pada nilai *gain* tertinggi bernilai 0.33725821 pada atribut JUMLAH_TANGGUNGAN. Atribut JUMLAH_TANGGUNGAN memiliki tiga nilai yaitu SEDIKIT, SEDANG, dan BANYAK. Nilai atribut BANYAK sudah mengklasifikasikan kasus menjadi satu yaitu dengan hasil MACET sehingga tidak perlu dilakukan perhitungan lebih lanjut. Pada atribut yang bernilai SEDIKIT terdiri dari kelas LANCAR berjumlah 17 dan kelas MACET berjumlah dua, data yang terletak pada kelas MACET merupakan *noise*, sehingga perlu dilakukan penghentian pembentukan *tree* lebih awal berdasarkan nilai minimum *threshold*, hal tersebut dikarenakan apabila diolah akan menghasilkan pohon dengan *subtree* yang panjang atau yang biasa disebut dengan *overfitting* (Liao, 2007). Sehingga, atribut bernilai SEDIKIT dapat disimpulkan telah mengklasifikasi kasus menjadi kelas LANCAR. Sedangkan pada nilai atribut SEDANG masih perlu dilakukan perhitungan lebih lanjut mengingat nilai atribut tersebut belum mengklasifikasi kasus menjadi satu keputusan. Dari perhitungan yang telah dilakukan, maka dihasilkan pohon keputusan yang tertera dalam Gambar 4.2.



Gambar 4. 2 Hasil *Tree* dari Perhitungan Pertama

Langkah 3: Mencari *node* cabang

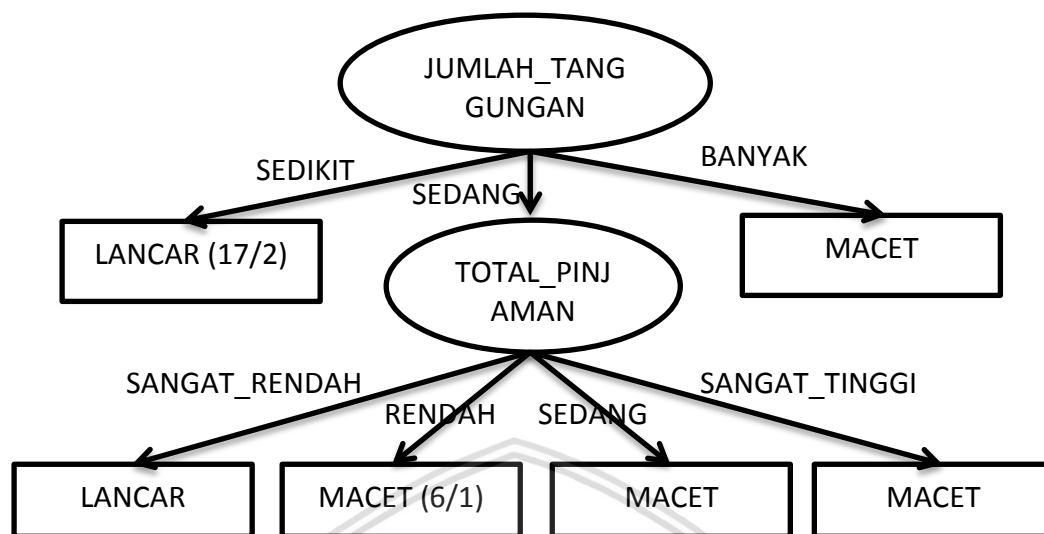
Perhitungan dilakukan untuk mencari *node* cabang dari nilai atribut JUMLAH_TANGGUNGAN adalah SEDANG. Atribut yang digunakan dalam perhitungan *node* cabang ini adalah semua atribut selain dari atribut JUMLAH_TANGGUNGAN yang telah menjadi *node* akar. Perhitungan dilakukan dengan cara menghitung jumlah kasus untuk hasil LANCAR dan MACET. Setelah itu, dilakukan perhitungan *entropy* dari semua kasus berdasarkan Persamaan 2.2 dan perhitungan *gain* untuk kemudian didapatkan nilai *gain* tertinggi berdasarkan Persamaan 2.1. Hasil dari perhitungan *entropy* dan *gain* dapat dilihat pada Tabel 4.24 dan dijelaskan lebih lengkap dalam Lampiran H.2.

Tabel 4. 24 Perhitungan Nilai *Entropy* dan *Gain* Tiap Atribut untuk Penentuan Node Cabang 1.1

Node	Atribut	Nilai	Entropi	Gain
1.1	JENIS_KELAMIN			0.0454357
		L	0.970951	
		P	0.650022	

Node	Atribut	Nilai	Entropi	Gain
	TOTAL_PINJAMAN			0.63718139
		SANGAT_RENDAH	0	
		RENDAH	0	
		SEDANG	0.591673	
		TINGGI	0	
		SANGAT_TINGGI	0	
	STATUS_PERNIKAHAN			0.11168754
		MENIKAH	0.836641	
		BELUM_MENIKAH	0	
		CERAI	0	
	PENDAPATAN_UTAMA			0.16113667
		RENDAH	0	
		SEDANG	0.979869	
		TINGGI	0	
	TOTAL_BIAYA_HIDUP			0.33855795
		RENDAH	0.991076	
		SEDANG	0	
		TINGGI	0	

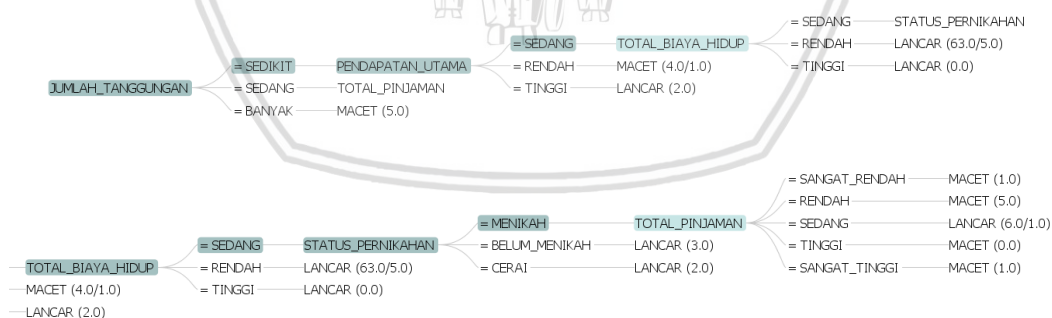
Pada Tabel 4.25 didapatkan hasil bahwa nilai *gain* tertinggi terdapat pada atribut TOTAL_PINJAMAN sebesar 0.63718139, sehingga atribut TOTAL_PINJAMAN menjadi *node* cabang dari atribut JUMLAH_TANGGUNGAN dengan nilai SEDANG. Atribut TOTAL_PINJAMAN mempunyai empat nilai yaitu SANGAT_RENDAH, RENDAH, SEDANG, dan SANGAT_TINGGI. Sedangkan nilai TINGGI dapat dieliminasi dikarenakan tidak terdapat kasus LANCAR maupun MACET pada perhitungan *node* cabang di atas. Nilai SANGAT_RENDAH, RENDAH, dan SANGAT_TINGGI sudah mengklasifikasikan kasus menjadi satu, sehingga tidak perlu dilakukan perhitungan lebih lanjut. Sedangkan pada nilai SEDANG terdapat kasus LANCAR berjumlah satu dan kasus MACET berjumlah enam, sehingga dilakukan penghentian pembentukan *tree* lebih awal yang menghasilkan klasifikasi kasus menjadi satu yaitu MACET. Dari perhitungan yang telah dilakukan, telah terbentuk *tree* yang sempurna. Hasil *tree* yang dihasilkan dapat dilihat dalam Gambar 4.3.



Gambar 4. 3 Hasil Tree yang Terbentuk

4.4.2 Learning Process Menggunakan Weka

Proses pembelajaran menggunakan *Weka* bertujuan untuk menghasilkan *rule* dalam bentuk pohon keputusan berdasarkan algoritme C4.5. Data yang digunakan pada tahap ini adalah keseluruhan data latih hasil dari pembagian data yang berjumlah 148 records. Algoritme C4.5 dapat digunakan dengan memilih fitur *J48* pada *Weka*. Secara otomatis *Weka* akan menghasilkan *tree* secara sempurna. Hasil dari *tree* yang dihasilkan dapat dilihat dalam Gambar 4.4 dan Gambar 4.5.



Gambar 4. 4 Hasil Tree dengan Atribut JUMLAH_TANGGUNGAN adalah SEDIKIT Menggunakan Weka



Gambar 4. 5 Hasil Tree dengan Atribut JUMLAH_TANGGUNGAN adalah SEDANG dan BANYAK Menggunakan Weka

Dari hasil *rule* yang dihasilkan dalam bentuk pohon keputusan seperti yang tertera dalam Gambar 4.4 dan 4.5 dapat dijelaskan sebagai berikut:

1. Jika jumlah tanggungan sedikit dan pendapatan utama sedang dan total biaya hidup sedang dan status pernikahan menikah dan total pinjaman sangat rendah, maka status pinjaman adalah macet.
2. Jika jumlah tanggungan sedikit dan pendapatan utama sedang dan total biaya hidup sedang dan status pernikahan menikah dan total pinjaman rendah, maka status pinjaman adalah macet.
3. Jika jumlah tanggungan sedikit dan pendapatan utama sedang dan total biaya hidup sedang dan status pernikahan menikah dan total pinjaman sedang, maka status pinjaman adalah lancar.
4. Jika jumlah tanggungan sedikit dan pendapatan utama sedang dan total biaya hidup sedang dan status pernikahan menikah dan total pinjaman tinggi, maka status pinjaman adalah macet.
5. Jika jumlah tanggungan sedikit dan pendapatan utama sedang dan total biaya hidup sedang dan status pernikahan menikah dan total pinjaman sangat tinggi, maka status pinjaman adalah macet.
6. Jika jumlah tanggungan sedikit dan pendapatan utama sedang dan total biaya hidup sedang dan status pernikahan belum menikah, maka status pinjaman adalah lancar.
7. Jika jumlah tanggungan sedikit dan pendapatan utama sedang dan total biaya hidup sedang dan status pernikahan cerai, maka status pinjaman adalah lancar.
8. Jika jumlah tanggungan sedikit dan pendapatan utama sedang dan total biaya hidup sedang dan status pernikahan belum menikah, maka status pinjaman adalah lancar.
9. Jika jumlah tanggungan sedikit dan pendapatan utama sedang dan total biaya hidup rendah, maka status pinjaman adalah lancar.
10. Jika jumlah tanggungan sedikit dan pendapatan utama sedang dan total biaya hidup tinggi, maka status pinjaman adalah lancar.
11. Jika jumlah tanggungan sedikit dan pendapatan utama rendah, maka status pinjaman adalah macet.
12. Jika jumlah tanggungan sedikit dan pendapatan utama tinggi, maka status pinjaman adalah macet.
13. Jika jumlah tanggungan sedang dan total pinjaman sangat rendah dan total biaya hidup sedang, maka status pinjaman adalah macet.

14. Jika jumlah tanggungan sedang dan total pinjaman sangat rendah dan total biaya hidup rendah, maka status pinjaman adalah lancar.
15. Jika jumlah tanggungan sedang dan total pinjaman sangat rendah dan total biaya hidup tinggi, maka status pinjaman adalah lancar.
16. Jika jumlah tanggungan sedang dan total pinjaman rendah, maka status pinjaman adalah macet.
17. Jika jumlah tanggungan sedang dan total pinjaman sedang, maka status pinjaman adalah macet.
18. Jika jumlah tanggungan sedang dan total pinjaman tinggi, maka status pinjaman adalah macet.
19. Jika jumlah tanggungan sedang dan total pinjaman sangat tinggi, maka status pinjaman adalah macet.
20. Jika jumlah tanggungan banyak, maka status pinjaman adalah macet.

4.5 Perancangan Sistem

Rancangan sistem merupakan cetakan untuk melakukan implementasi (kodifikasi). Dalam hal ini, perancangan yang dilakukan berbasis *object* dengan memanfaatkan bahasa perancangan UML (*Unified Modeling Language*) dan divisualisasikan dengan *sequence diagram* dan *class diagram*.

4.5.1 Sequence Diagram

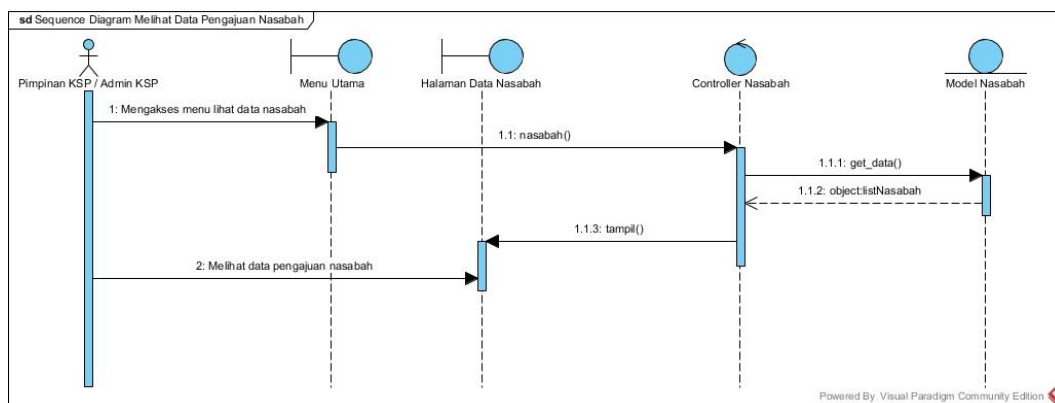
Sequence Diagram menjelaskan tentang bagaimana antar komponen berinteraksi menggunakan mekanisme pertukaran pesan. Berikut beberapa *sequence diagram* untuk sistem rekomendasi pengajuan kredit yang dibangun:

4.5.1.1 Sequence Diagram Memasukkan Dataset (SD-SRPK-1)

Alur mekanisme interaksi sekuensial antar aktor Pimpinan KSP / Admin KSP, dengan *controller dataset* yang menangani permintaan aktor Pimpinan KSP / Admin KSP untuk memasukkan *dataset*, *library csvimport* untuk menangani pembacaan *file csv*, dengan *boundary* menu utama untuk mengakses menu utama, *boundary* halaman input *dataset* untuk menampilkan *form input dataset*, *boundary warning success* dan *warning failed* untuk menampilkan *pop-up warning success* atau *failed*, serta *model* nasabah, transaksi, *datatrain*, dan *datatest* untuk menyimpan data ditampilkan dalam Lampiran B.2. Dan terdapat juga *Simple CLI Weka* untuk penggunaan beberapa fungsi terkait dengan klasifikasi menggunakan C4.5.

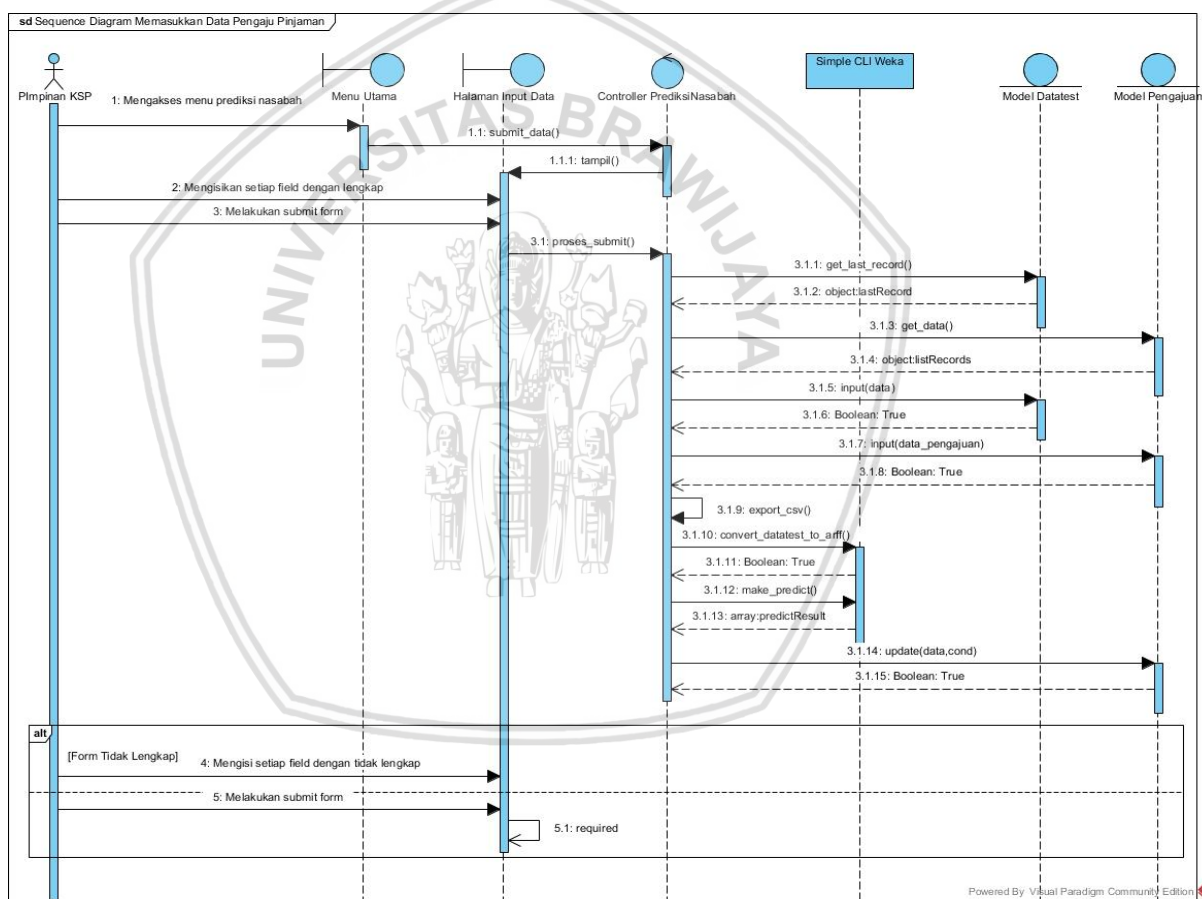
4.5.1.2 Sequence Diagram Melihat Data Pengajuan Nasabah (SD-SRPK-2)

Gambar 4.6 menjelaskan alur mekanisme interaksi sekuensial antar aktor Pimpinan KSP / Admin KSP, dengan *controller* nasabah yang menangani permintaan aktor Pimpinan KSP / Admin KSP, dengan *boundary* menu utama untuk mengakses menu utama dan *boundary* halaman data pengajuan nasabah untuk menampilkan data pengajuan nasabah, serta *model* nasabah untuk mendapatkan data nasabah beserta dengan data transaksi.



Gambar 4. 6 Sequence Diagram Melihat Data Pengajuan Nasabah

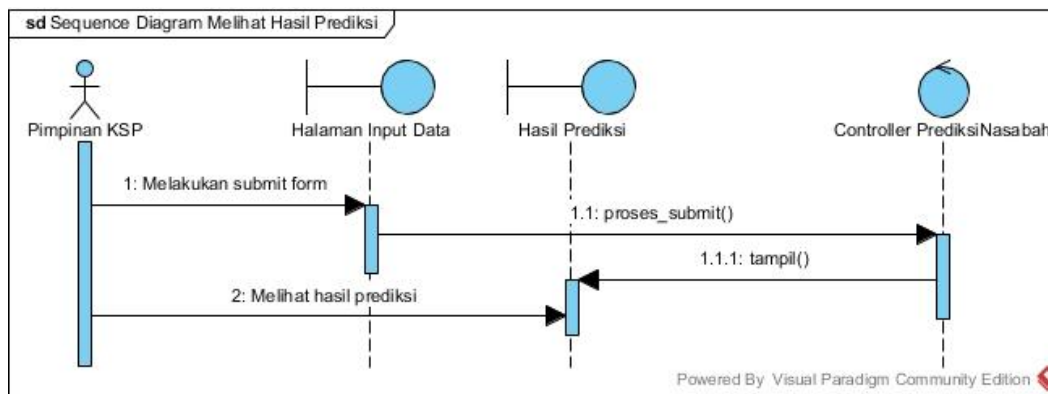
4.5.1.3 Sequence Diagram Memasukkan Data Pengaju Pinjaman (SD-SRPK-3)



Gambar 4. 7 Sequence Diagram Memasukkan Data Pengaju Pinjaman

Gambar 4.7 menjelaskan alur mekanisme interaksi sekuensial antar aktor Pimpinan KSP, dengan *controller* PrediksiNasabah yang menangani permintaan aktor Pimpinan KSP, dengan *boundary* menu utama untuk mengakses menu utama dan *boundary* halaman *input data* untuk menampilkan *form input data*, *model datatest* dan pengajuan, serta *Simple CLI Weka* untuk menangani beberapa fitur yang terkait dengan klasifikasi dengan algoritme C4.5.

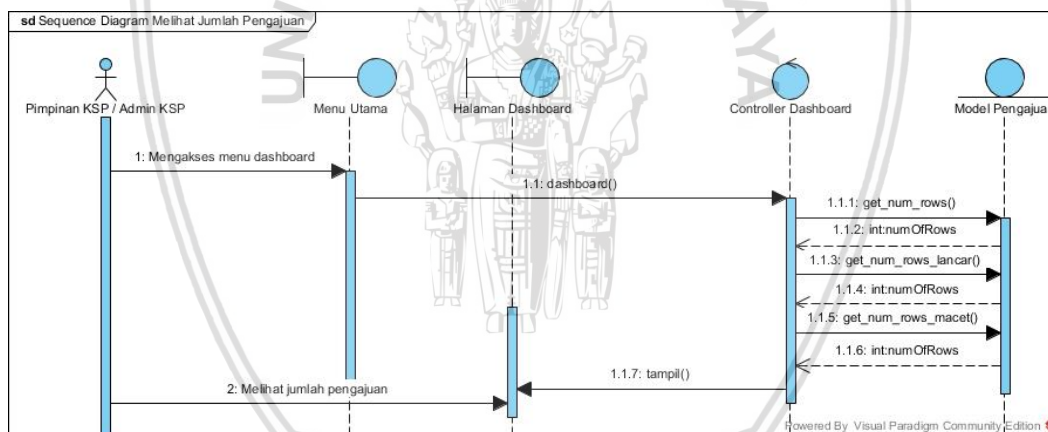
4.5.1.4 Sequence Diagram Melihat Hasil Prediksi (SD-SRPK-4)



Gambar 4. 8 Sequence Diagram Melihat Hasil Prediksi

Gambar 4.8 menjelaskan alur mekanisme interaksi sekuensial antar aktor Pimpinan KSP, dengan *controller* PrediksiNasabah yang menangani permintaan aktor Pimpinan KSP, dengan *boundary* halaman *input data* untuk menampilkan *form input data* serta melakukan *submit form* dan *boundary* hasil prediksi untuk menampilkan hasil prediksi yang didapat.

4.5.1.5 Sequence Diagram Melihat Jumlah Pengajuan (SD-SRPK-5)



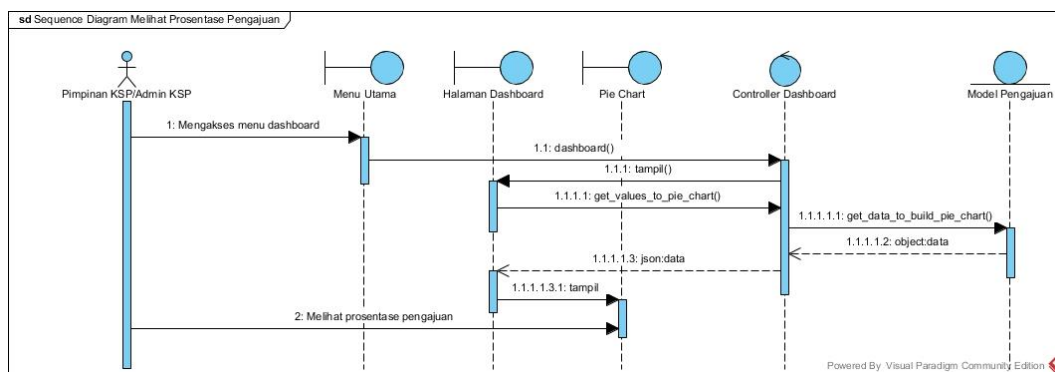
Gambar 4. 9 Sequence Diagram Melihat Jumlah Pengajuan

Gambar 4.9 menjelaskan alur mekanisme interaksi sekuensial antar aktor Pimpinan KSP / Admin KSP, dengan *controller dashboard* yang menangani permintaan aktor Pimpinan KSP / Admin KSP, dengan *boundary* menu utama untuk mengakses menu utama dan *boundary* halaman *dashboard* untuk menampilkan halaman *dashboard* beserta dengan jumlah pengajuan, serta *model* pengajuan untuk memperoleh data dari *database*.

4.5.1.6 Sequence Diagram Melihat Persentase Pengajuan (SD-SRPK-6)

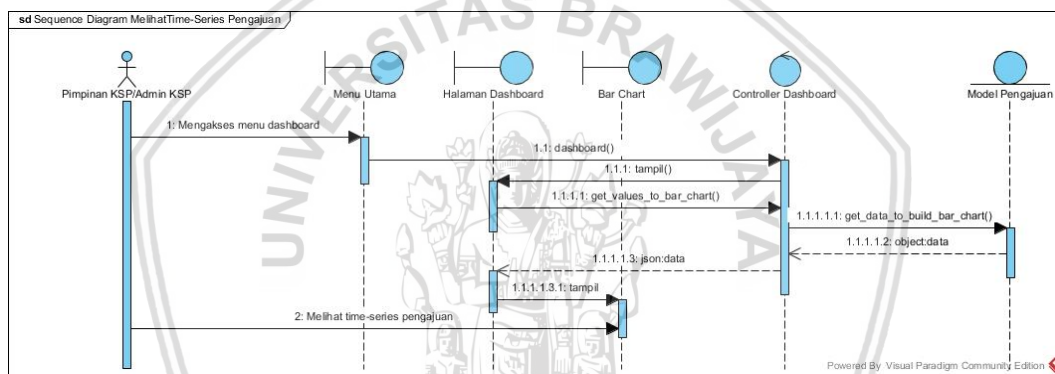
Gambar 4.10 menjelaskan alur mekanisme interaksi sekuensial antar aktor Pimpinan KSP / Admin KSP, dengan *controller dashboard* yang menangani permintaan aktor Pimpinan KSP / Admin KSP, dengan *boundary* menu utama untuk mengakses menu utama, *boundary* halaman *dashboard* untuk

menampilkan halaman *dashboard*, dan *boundary pie chart* untuk menampilkan data persentase pengajuan dalam bentuk *pie chart*, serta *model* pengajuan untuk memperoleh data dari *database*.



Gambar 4. 10 Sequence Diagram Melihat Persentase Pengajuan

4.5.1.7 Sequence Diagram Melihat Time-Series Pengajuan (SD-SRPK-7)

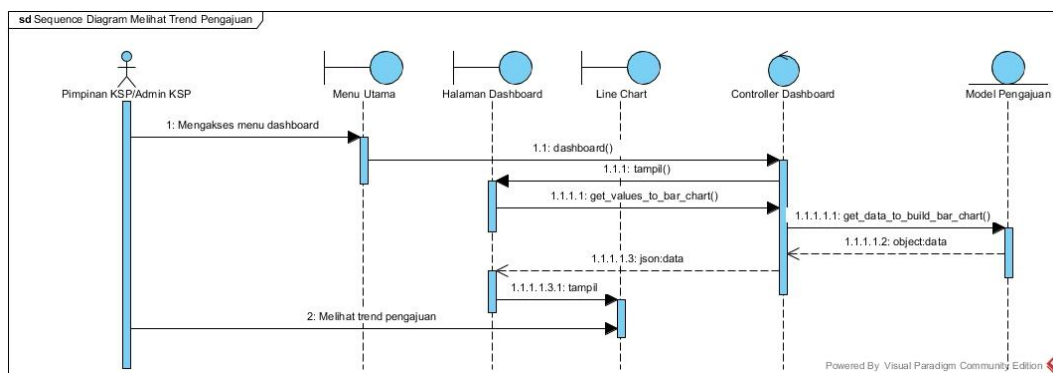


Gambar 4. 11 Sequence Diagram Melihat Time-Series Pengajuan

Gambar 4.11 menjelaskan alur mekanisme interaksi sekuensial antar aktor Pimpinan KSP / Admin KSP, dengan *controller dashboard* yang menangani permintaan aktor Pimpinan KSP / Admin KSP, dengan *boundary* menu utama untuk mengakses menu utama, *boundary* halaman *dashboard* untuk menampilkan halaman *dashboard*, dan *boundary bar chart* untuk menampilkan *time-series* pengajuan dalam bentuk *bar chart*, serta *model* pengajuan untuk memperoleh data dari *database*.

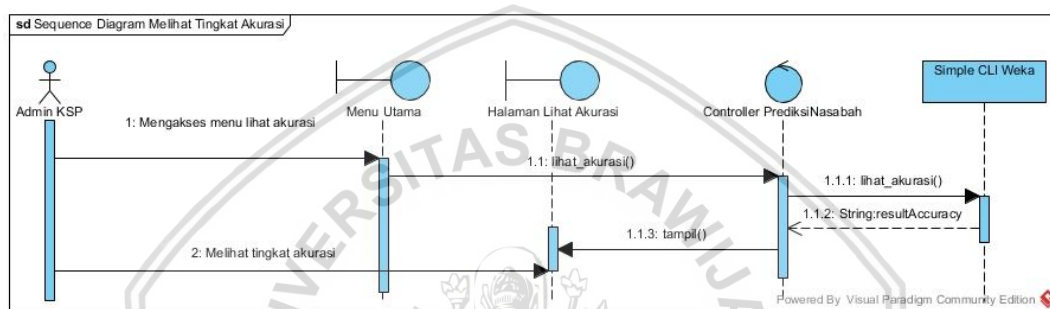
4.5.1.8 Sequence Diagram Melihat Trend Pengajuan (SD-SRPK-8)

Gambar 4.12 menjelaskan alur mekanisme interaksi sekuensial antar aktor Pimpinan KSP / Admin KSP, dengan *controller dashboard* yang menangani permintaan aktor Pimpinan KSP / Admin KSP, dengan *boundary* menu utama untuk mengakses menu utama, *boundary* halaman *dashboard* untuk menampilkan halaman *dashboard*, dan *boundary line chart* untuk menampilkan *time-series* pengajuan dalam bentuk *line chart*, serta *model* pengajuan untuk memperoleh data dari *database*.



Gambar 4. 12 Sequence Diagram Melihat Trend Pengajuan

4.5.1.9 Sequence Diagram Melihat Tingkat Akurasi (SD-SRPK-9)

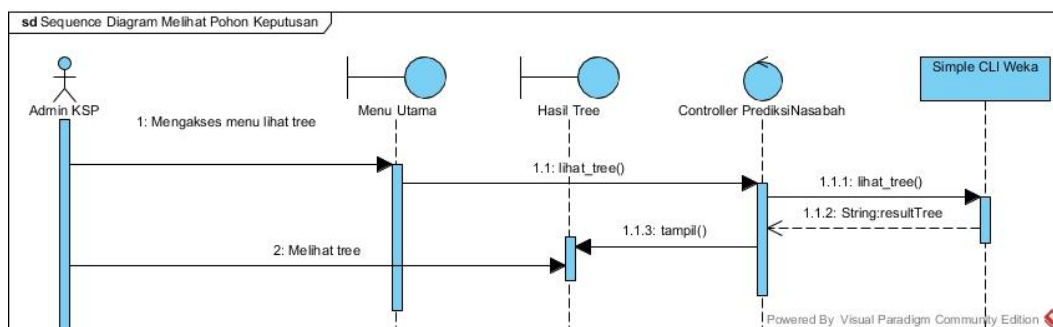


Gambar 4. 13 Sequence Diagram Melihat Tingkat Akurasi

Gambar 4.13 menjelaskan alur mekanisme interaksi sekuensial antar aktor Admin KSP, dengan *controller* PrediksiNasabah yang menangani permintaan aktor Admin KSP, dengan *boundary* menu utama untuk mengakses menu utama dan *boundary* halaman lihat akurasi untuk menampilkan akurasi dari sistem, serta *Simple CLI Weka* yang dapat digunakan untuk melakukan fungsi klasifikasi menggunakan algoritme C4.5.

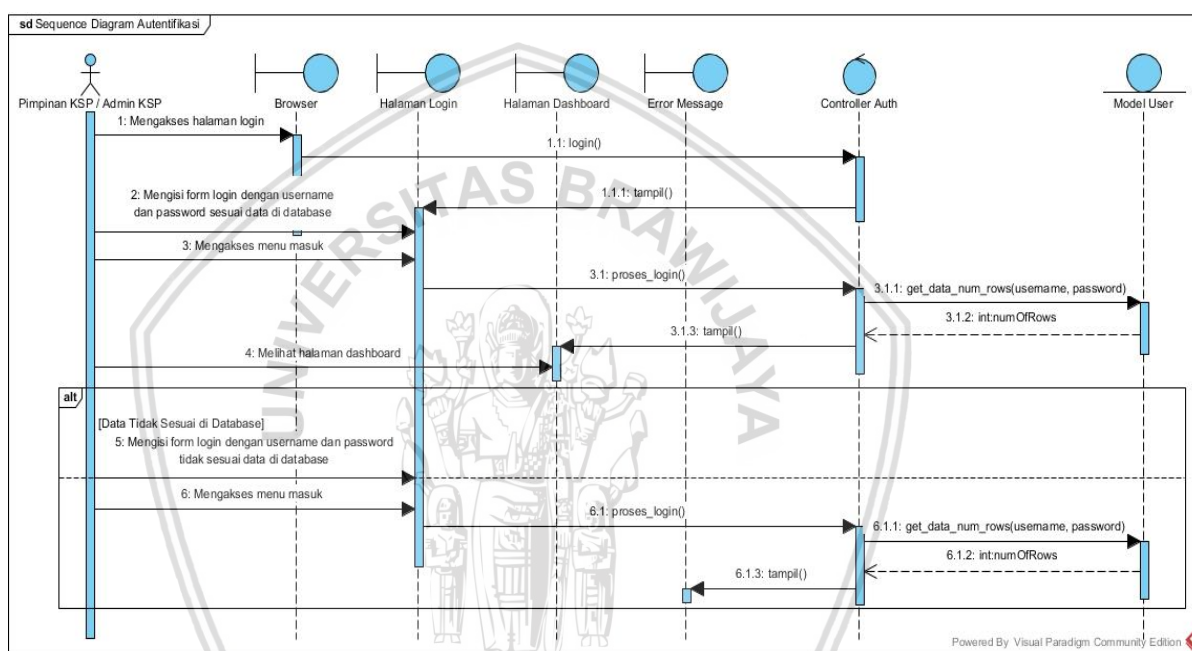
4.5.1.10 Sequence Diagram Melihat Pohon Keputusan (SD-SRPK-10)

Gambar 4.14 menjelaskan alur mekanisme interaksi sekuensial antar aktor Admin KSP, dengan *controller* PrediksiNasabah yang menangani permintaan aktor Admin KSP, dengan *boundary* menu utama untuk mengakses menu utama dan *boundary* hasil *tree* untuk menampilkan hasil *tree* yang terbentuk serta *Simple CLI Weka* yang dapat digunakan untuk melakukan fungsi klasifikasi menggunakan algoritme C4.5.



Gambar 4. 14 Sequence Diagram Melihat Pohon Keputusan

4.5.1.11 Sequence Diagram Autentifikasi (SD-SRPK-11)

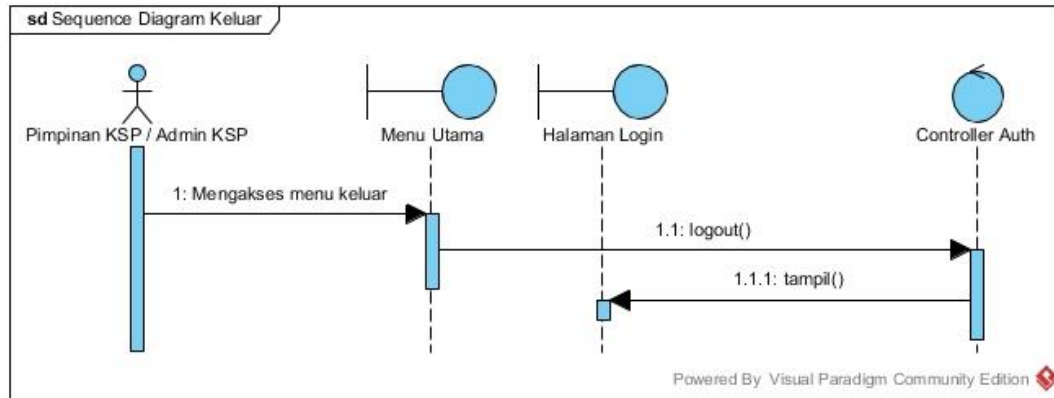


Gambar 4. 15 Sequence Diagram Autentifikasi

Gambar 4.15 menjelaskan alur mekanisme interaksi sekuensial antar aktor Pimpinan KSP / Admin KSP, dengan *controller Auth* yang menangani permintaan aktor Pimpinan KSP / Admin KSP, dengan *boundary browser* untuk mengakses sistem, *boundary halaman login* untuk menampilkan form *login*, *boundary halaman dashboard* untuk menampilkan halaman *dashboard* apabila Pimpinan KSP / Admin KSP berhasil *login*, serta *model user* untuk mendapatkan data *user*.

4.5.1.12 Sequence Diagram Keluar (SD-SRPK-12)

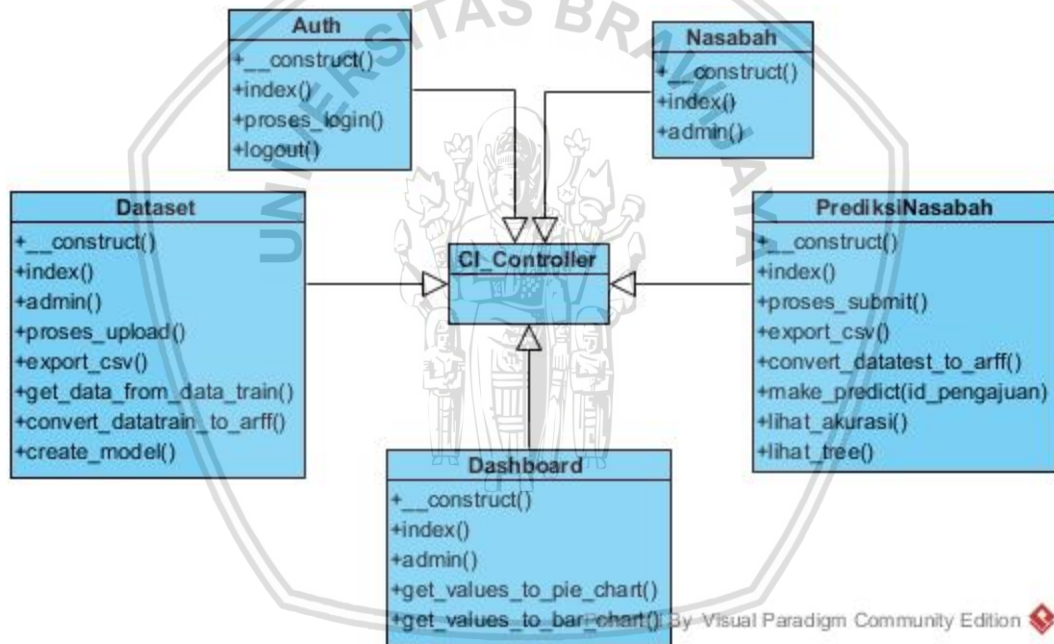
Gambar 4.16 menjelaskan alur mekanisme interaksi sekuensial antar aktor Pimpinan KSP / Admin KSP, dengan *controller Auth* yang menangani permintaan aktor Pimpinan KSP / Admin KSP, dengan *boundary menu utama* untuk mengakses menu keluar dan *boundary halaman login* untuk menampilkan *form login* apabila Pimpinan KSP / Admin KSP berhasil keluar.



Gambar 4. 16 Sequence Diagram Keluar

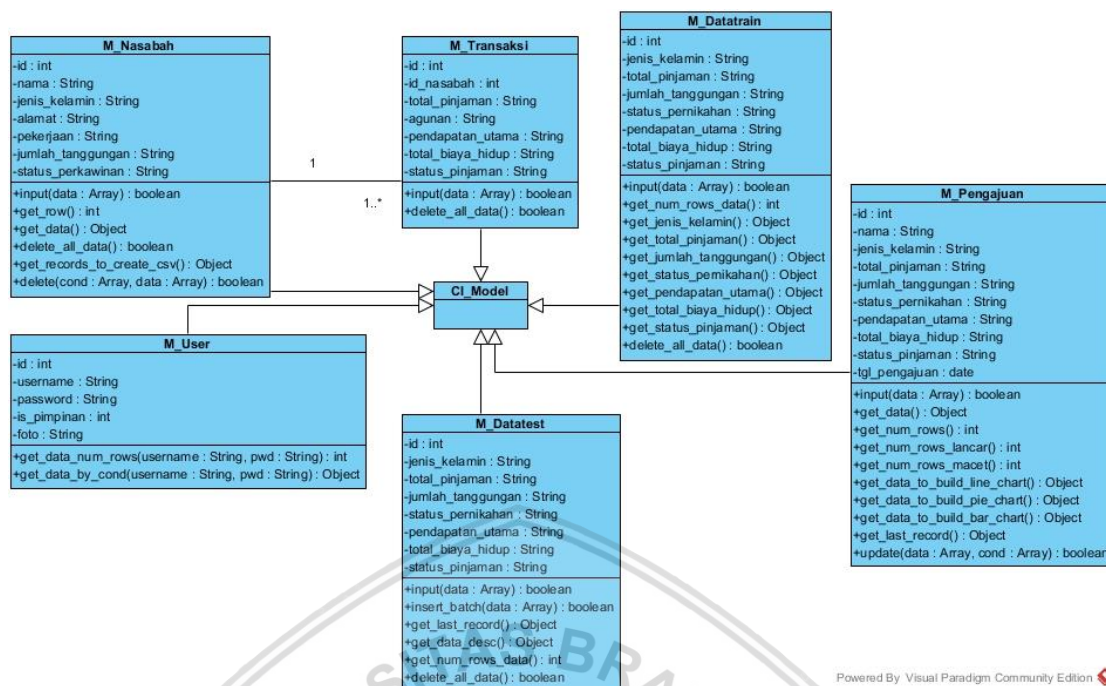
4.5.2 Class Diagram

Class Diagram adalah model statis yang menggambarkan struktur, deskripsi, dan relasi antar kelas dari sistem informasi yang dibangun.



Gambar 4. 17 Class Diagram sebagai logical class

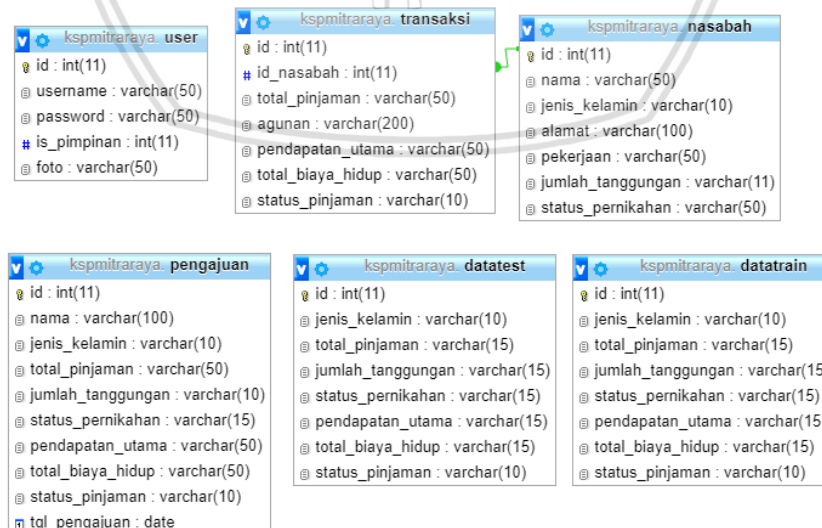
Gambar 4.17 merupakan representasi *logical class* dari pengolahan logika untuk pemanggilan kelas data dan tampilan. Terdapat lima kelas yaitu kelas *Auth*, *Nasabah*, *Dataset*, *Dashboard*, dan *PrediksiNasabah* yang menurun dari kelas *CI_Controller* dalam implementasi menggunakan *framework codeigniter*.



Gambar 4. 18 Class Diagram sebagai domain model

Gambar 4.18 menggambarkan kelas model dan relasi diantaranya, sehingga nantinya digunakan untuk acuan pemodelan data. Terdapat enam kelas yaitu M_Nasabah, M_User, M_Transaksi, M_Dataset, M_Pengajuan, dan M_Datatrain yang menuruni kelas CI_Model dalam implementasinya menggunakan framework codeigniter.

Pemodelan data didasarkan pada *class diagram* sebagai *domain model* dalam Gambar 4.18 dengan menggunakan *Physical Data Modelling* (PDM). Adapun model dari pemodelan data tertera dalam Gambar 4.19.

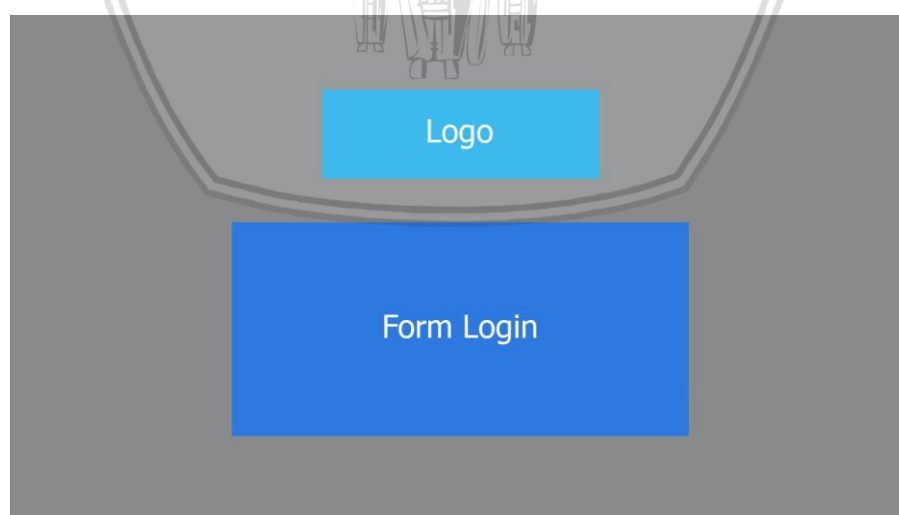


Gambar 4. 19 Physical Data Modelling

Dalam Gambar 4.19, dapat dilihat bahwa terdapat enam tabel yang merupakan representasi dari hasil pemodelan data menggunakan PDM. Enam tabel tersebut adalah tabel *user*, transaksi, nasabah, pengajuan, *datatest*, dan juga *datatrain*. Tabel *user* berfungsi untuk menyimpan data penggunaan sistem yang terdiri dari lima *field* yaitu *id* yang berperan sebagai *primary key*, *username*, *password*, *is_pimpinan*, dan foto. Tabel transaksi berfungsi untuk menyimpan data transaksi dari nasabah, terdiri dari tujuh *field* yaitu *id* yang berperan sebagai *primary key*, *id_nasabah* yang berperan sebagai *foreign key* dan mereferensi ke *field id* pada tabel nasabah. Tabel nasabah berfungsi untuk menyimpan data nasabah, terdiri dari tujuh *field* yaitu *id* yang berperan sebagai *primary key*, nama, jenis_kelamin, alamat, pekerjaan, jumlah_tanggungan, dan status_pernikahan. Tabel pengajuan berfungsi untuk menyimpan data pengajuan yang telah dilakukan pada KSP Mitra Raya Wates, terdiri dari 10 *field* yaitu *id* yang berperan sebagai *primary key*, nama, jenis_kelamin, total_pinjaman, jumlah_tanggungan, status_pernikahan, pendapatan_utama, total_biaya_hidup, status_pinjaman, dan tgl_pengajuan. Tabel *datatrain* berfungsi untuk menyimpan data yang akan digunakan sebagai *train set*, terdiri dari delapan *field*, yaitu *id* sebagai *primary key*, jenis_kelamin, total_pinjaman, jumlah_tanggungan, status_pernikahan, pendapatan_utama, total_biaya_hidup, dan status_pinjaman. Dan Tabel *datatest* berfungsi untuk menyimpan data yang akan dijadikan sebagai *testing set*, terdiri dari delapan *field*, yaitu *id* sebagai *primary key*, jenis_kelamin, total_pinjaman, jumlah_tanggungan, status_pernikahan, pendapatan_utama, total_biaya_hidup, dan status_pinjaman.

4.6 Perancangan Visualisasi *Dashboard*

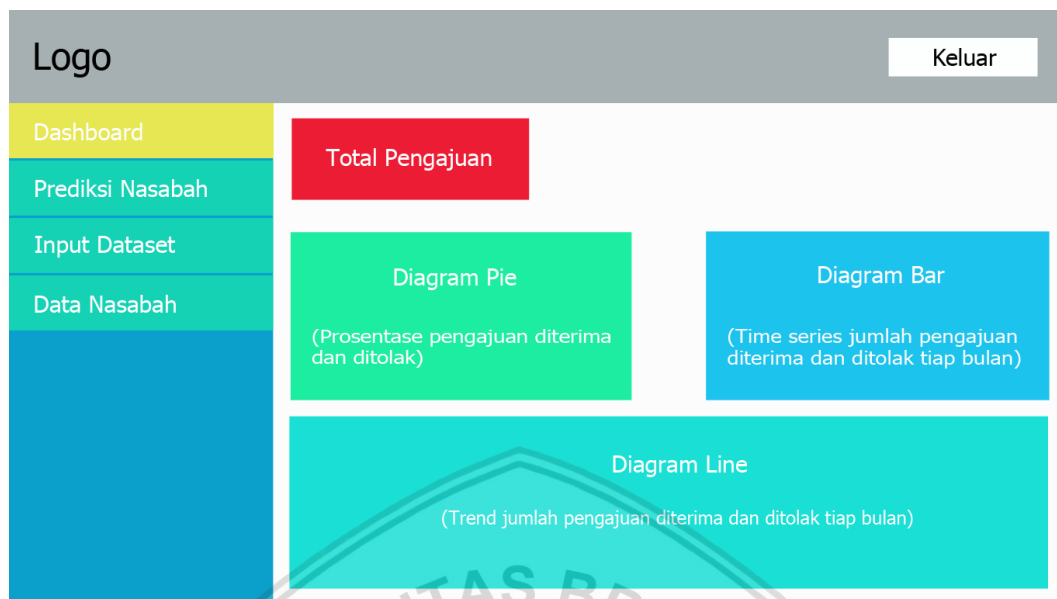
4.6.1 Perancangan Visualisasi Halaman *Login*



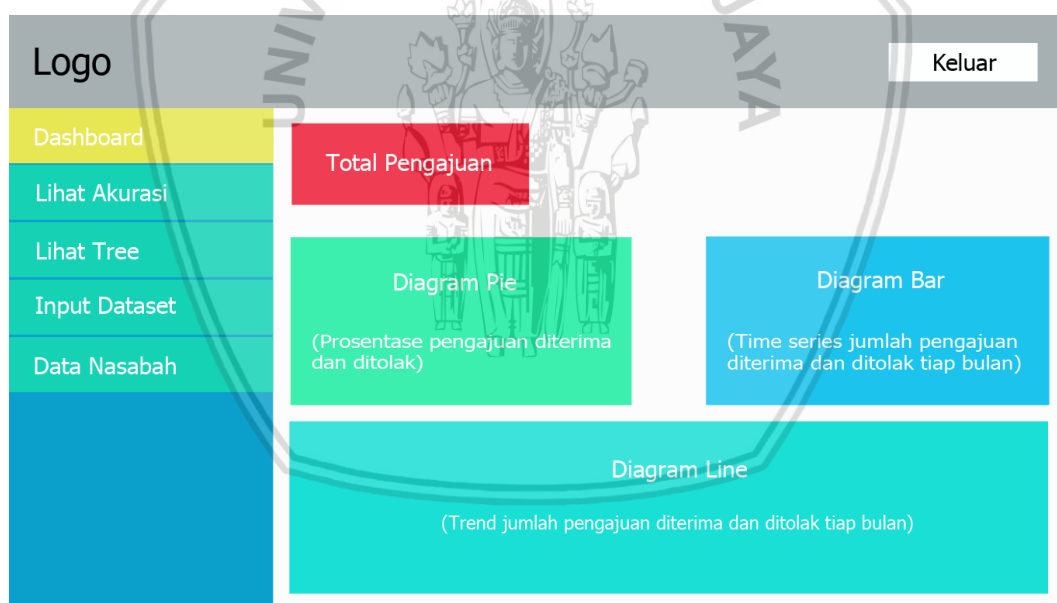
Gambar 4. 20 Perancangan Visualisasi Halaman *Login*

Gambar 4.22 menggambarkan halaman *login*. Pada halaman ini dimuat logo dari KSP Mitra Raya Wates dan juga *form login* untuk *login* bagi Pimpinan KSP maupun Admin KSP.

4.6.2 Perancangan Visualisasi Halaman *Dashboard*



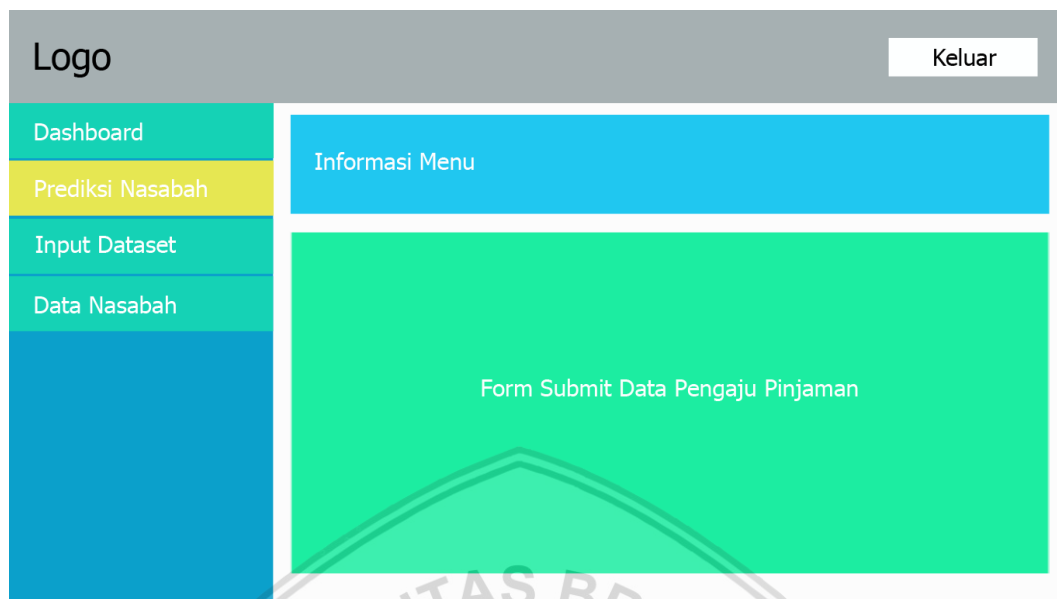
Gambar 4. 21 Perancangan Visualisasi Halaman *Dashboard* Pimpinan KSP



Gambar 4. 22 Perancangan Visualisasi Halaman *Dashboard* Admin KSP

Gambar 4.23 menggambarkan halaman *dashboard* untuk aktor Pimpinan KSP dan Gambar 4.24 menggambarkan halaman *dashboard* untuk aktor Admin KSP. Pada halaman ini dimuat total pengajuan yang telah dilakukan baik untuk total pengajuan yang diterima maupun yang ditolak, *diagram pie* sebagai representasi persentase pengajuan diterima dan ditolak, *diagram bar* yang merepresentasikan *time-series* jumlah pengajuan diterima dan ditolak, dan *diagram line* yang merepresentasikan *trend* jumlah pengajuan diterima dan ditolak tiap bulan pada KSP Mitra Raya Wates.

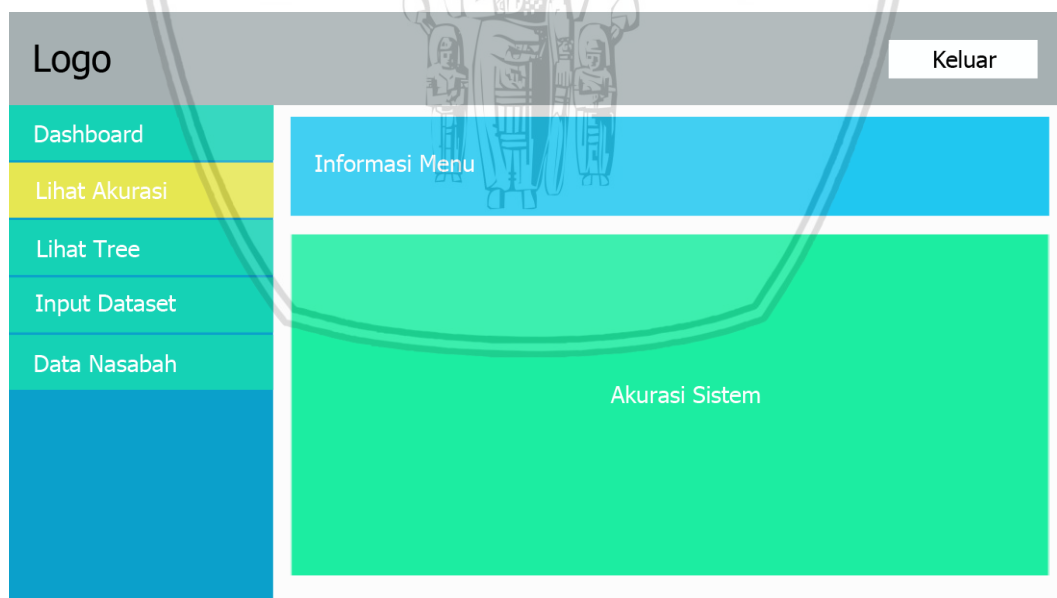
4.6.3 Perancangan Visualisasi Halaman Prediksi Nasabah



Gambar 4. 23 Perancangan Visualisasi Halaman Prediksi Nasabah

Gambar 4.25 menggambarkan halaman prediksi nasabah. Pada halaman ini dimuat *form submit data* pengaju pinjaman yang dapat digunakan Pimpinan KSP untuk memasukkan data nasabah yang ingin dilihat prediksi pengajuan kreditnya.

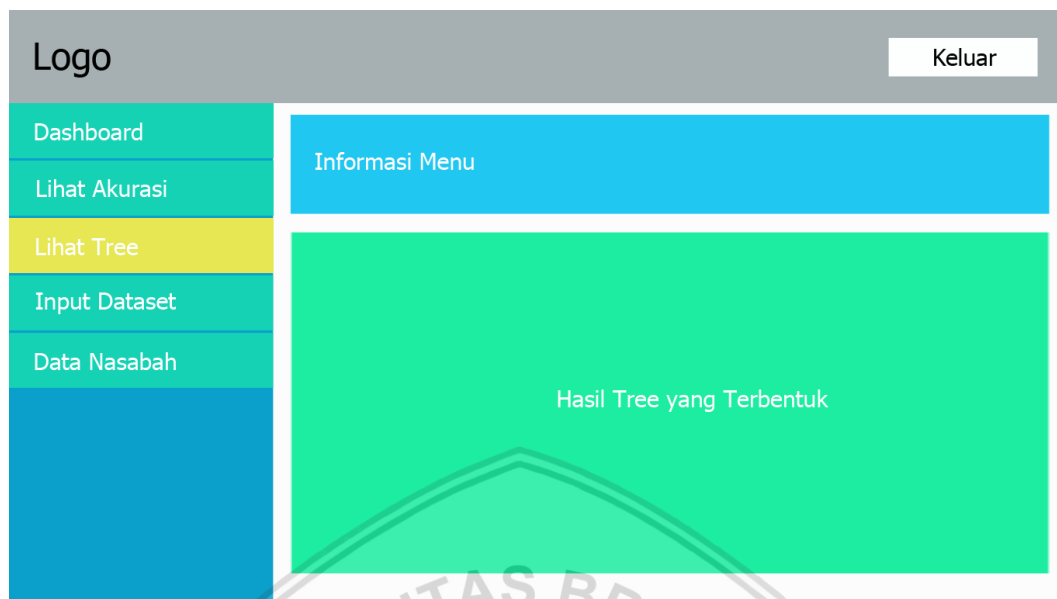
4.6.4 Perancangan Visualisasi Halaman Lihat Akurasi



Gambar 4. 24 Perancangan Visualisasi Halaman Lihat Akurasi

Gambar 4.26 menggambarkan halaman lihat akurasi. Pada halaman ini dimuat mengenai informasi menu yang nantinya akan memudahkan Admin KSP untuk menggunakan menu lihat akurasi dan juga informasi mengenai akurasi sistem yang dibuat.

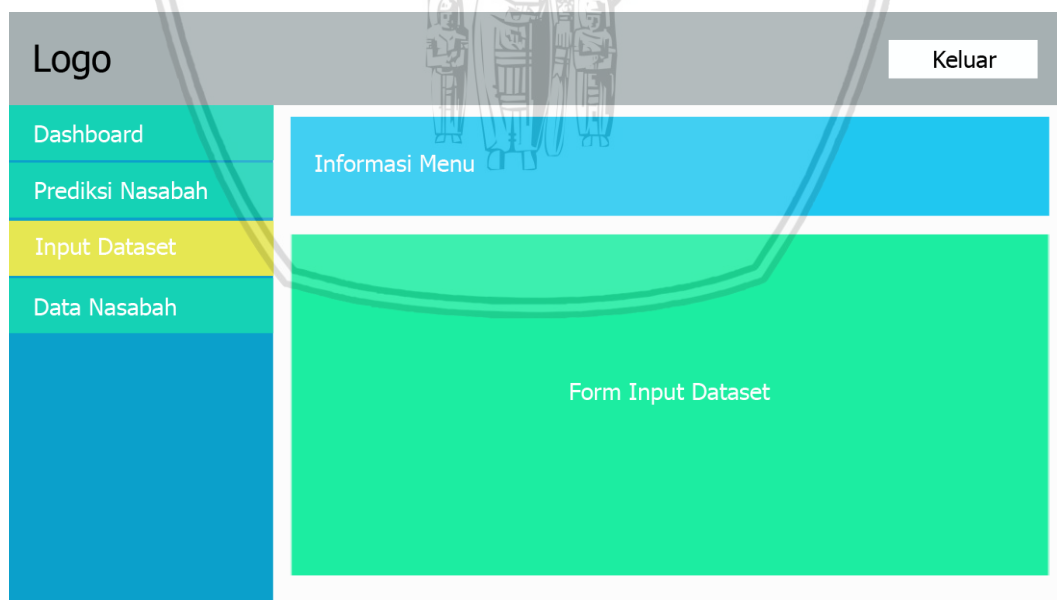
4.6.5 Perancangan Visualisasi Halaman Lihat Tree



Gambar 4. 25 Perancangan Visualisasi Halaman Lihat Tree

Gambar 4.27 menggambarkan halaman lihat *tree*. Pada halaman ini dimuat mengenai informasi menu yang nantinya akan memudahkan Admin KSP untuk menggunakan menu lihat *tree* dan juga hasil *tree* yang terbentuk dari sistem yang dibuat.

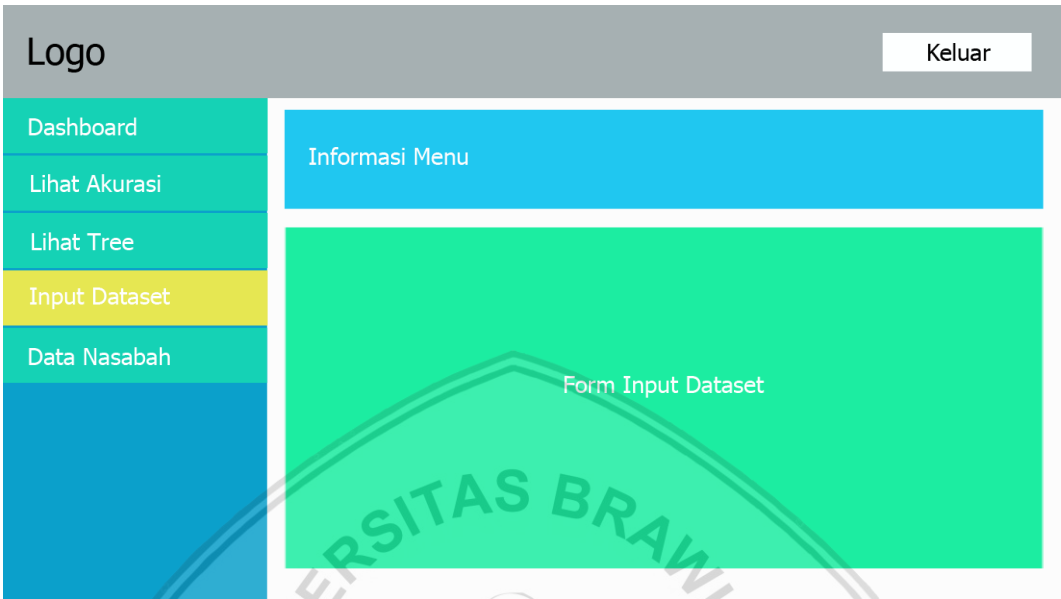
4.6.6 Perancangan Visualisasi Halaman Input Dataset



Gambar 4. 26 Perancangan Visualisasi Halaman Input Dataset Pimpinan KSP

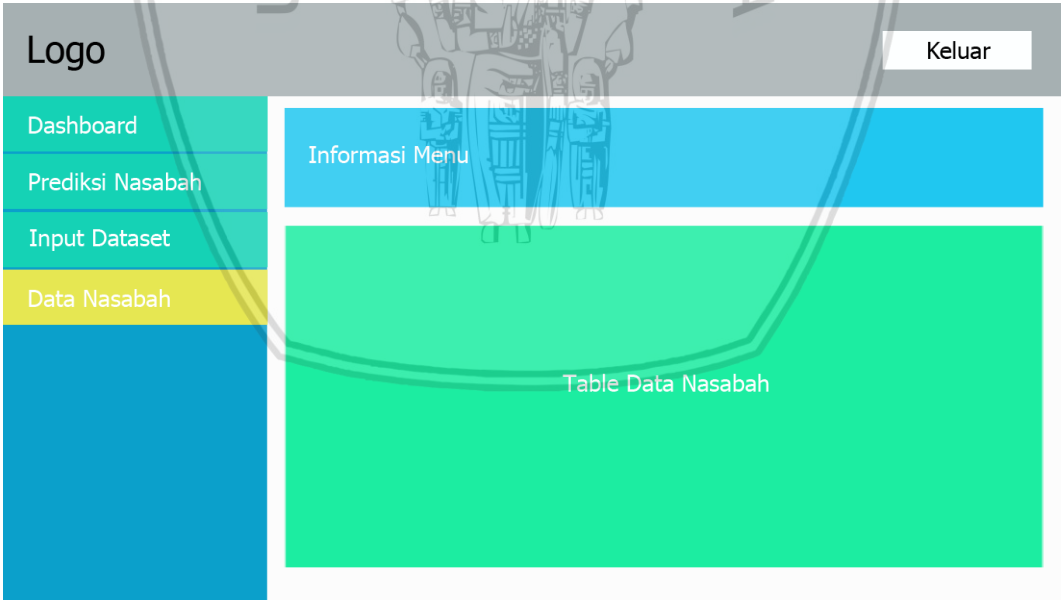
Gambar 4.28 menggambarkan halaman lihat *input dataset* untuk aktor Pimpinan KSP dan Gambar 4.29 menggambarkan halaman *input dataset* untuk Admin KSP. Pada halaman ini dimuat mengenai informasi menu yang nantinya

akan memudahkan Pimpinan KSP dan Admin KSP untuk menggunakan menu *input dataset* dan juga *form* untuk melakukan pengunggahan *dataset* ke sistem.

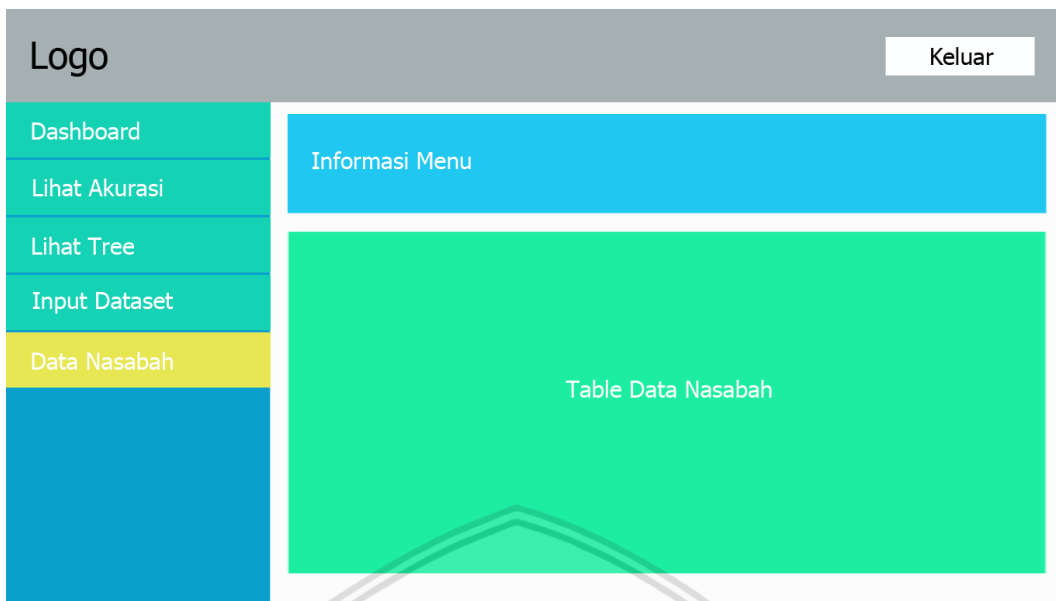


Gambar 4. 27 Perancangan Visualisasi Halaman *Input Dataset* Admin KSP

4.6.7 Perancangan Visualisasi Halaman Data Pengajuan Nasabah



Gambar 4. 28 Perancangan Visualisasi Halaman Lihat Data Pengajuan Nasabah
Pimpinan KSP



Gambar 4. 29 Perancangan Visualisasi Halaman Lihat Data Pengajuan Nasabah Admin KSP

Gambar 4.30 menggambarkan halaman lihat data pengajuan nasabah untuk aktor Pimpinan KSP dan Gambar 4.31 menggambarkan halaman lihat data pengajuan nasabah untuk aktor Admin KSP. Pada halaman ini dimuat mengenai informasi menu yang nantinya akan memudahkan Pimpinan KSP dan Admin KSP untuk melihat data pengajuan nasabah dan juga tabel yang berisi data pengajuan nasabah beserta dengan preferensinya berdasarkan data yang telah diunggah di *database*.

BAB 5 IMPLEMENTASI

Bab ini akan menjelaskan bagaimana hasil dari perancangan yang telah dilakukan akan diimplementasikan dalam bentuk sistem yang dapat berfungsi secara utuh. Bab ini dibagi menjadi beberapa bagian diantaranya adalah spesifikasi sistem, batasan implementasi, implementasi sistem, dan implementasi visualisasi.

5.1 Spesifikasi Sistem

Spesifikasi sistem untuk implementasi sistem yang akan dibangun ini dibagi menjadi dua spesifikasi, yaitu spesifikasi perangkat keras dan spesifikasi perangkat lunak.

5.1.1 Spesifikasi Perangkat Keras

Ruang lingkup perangkat keras minimum yang digunakan untuk membangun sistem ini terspesifikasikan pada Tabel 5.1.

Tabel 5. 1 Spesifikasi Minimum Perangkat Keras

Nama	Spesifikasi
Perangkat Keras	Komputer/Laptop
Prosesor	Pentium 1
Kapasitas RAM	128 MB
Kapasitas Harddisk	400 MB
Kartu Grafis	64 MB

Pada Tabel 5.1 dapat diketahui bahwa minimum spesifikasi perangkat keras yang dapat digunakan untuk membangun sistem ini adalah perangkat keras komputer atau laptop, proses minimum pentium 1, kapasitas RAM minimum 128 MB, kapasitas Harddisk minimum 400 MB, dan kartu grafis minimum 64 MB.

5.1.2 Spesifikasi Perangkat Lunak

Sedangkan spesifikasi perangkat lunak yang dapat digunakan untuk membantu dalam proses pembangunan sistem terspesifikasikan pada Tabel 5.2.

Tabel 5. 2 Spesifikasi Perangkat Lunak

Nama	Spesifikasi
Sistem Operasi	<i>Windows 8.1</i>
Web Browser	<i>Google Chrome</i>
Bahasa Pemrograman	PHP, HTML, <i>Javascript</i> , CSS, SQL
Framework	<i>CodeIgniter 3.1.6</i>

Nama	Spesifikasi
Web Server	<i>Apache MySQL</i>
Tool Tambahan	<i>Weka 3.8</i>

Pada Tabel 5.2 dapat diketahui spesifikasi perangkat lunak yang dapat digunakan untuk membantu dalam proses pembangunan sistem ini, diantaranya adalah sistem operasi *windows 8.1*, *web browser google chrome*, bahasa pemrograman yang digunakan adalah *PHP, HTML, Javascript, CSS, dan SQL. Framework* yang dapat digunakan adalah *CodeIgniter 3.1.6* dengan web servernya adalah *Apache Mysql*. Dan *tool* tambahan adalah *Weka 3.8*.

5.2 Batasan Implementasi

Dalam membangun sistem ini, terdapat batasan-batasan yang peneliti gunakan, diantaranya adalah:

1. Sistem yang dikembangkan berbasis web.
2. Algoritme klasifikasi yang digunakan adalah C4.5.
3. Sistem hanya pada ruang lingkup KSP Mitra Raya.
4. Masukan sistem berupa preferensi nasabah yang berpengaruh terhadap macet tidaknya seorang nasabah, berikut preferensinya : jenis kelamin, total pinjaman, jumlah tanggungan, status pernikahan, pendapatan utama, total biaya hidup, status pinjaman.
5. Luaran sistem berupa prediksi kredit macet atau lancar dari pengajuan kredit nasabah.

5.3 Implementasi Sistem

Implementasi sistem merupakan tahap dimana hasil dari perancangan berupa diagram-diagram UML dikonversikan menjadi bentuk bahasa pemrograman. Untuk membangun sistem ini, maka dibutuhkan beberapa fungsi diantaranya adalah fungsi untuk *input dataset*, prediksi nasabah, lihat akurasi, dan lihat *tree*.

5.3.1 Implementasi Fungsi *Input Dataset*

Fungsi *input dataset* terdapat pada kelas *Controller Dataset*, yang mana merupakan fungsi untuk memasukkan *dataset* ke *database*. Nantinya, *dataset* yang telah diunggah ke *database* akan bertindak sebagai *data training* sehingga proses prediksi dapat dilakukan oleh sistem.

<code>public function proses_upload()</code>	
1	<code>if (file_exists("../assets/uploads/upload_datatrain.csv")) {</code>
2	<code> unlink("../assets/uploads/upload_datatrain.csv");</code>
3	<code>}</code>
4	<code>\$name_file = "upload_datatrain";</code>

```

5  $config['upload_path'] = './assets/uploads/';
6  $config['allowed_types'] = 'csv';
7  $config['max_size'] = '2048';
8  $config['file_name'] = $name_file;
9
10 $this->load->library('upload', $config);
11
12 $this->upload->initialize($config);
13 if ($this->upload->do_upload('dataset')) {
14     $file_upload = $this->upload->data();
15     $file_path = './assets/uploads/'.$file_upload['file_name'];
16
17     if ($this->csvimport->get_array($file_path)) {
18         $csv_array = $this->csvimport->get_array($file_path);
19         $count_row = $this->M_Nasabah->get_row();
20         if ($count_row > 0) {
21             $this->M_Nasabah->delete_all_data();
22             $this->M_Transaksi->delete_all_data();
23         }
24         foreach ($csv_array as $key => $value) {
25             $data_nasabah[] = array('id' => $value['No'], 'nama'
=>
26             $value['Nama_Anggota'], 'jenis_kelamin' =>
27             $value['Jenis_Kelamin'], 'alamat' => $value['Alamat'], 'pekerjaan' =>
28             $value['Pekerjaan'], 'jumlah_tanggungan' =>
29             $value['Jumlah_Tanggungan'], 'status_pernikahan' =>
30             $value['Status_Pernikahan']);
31
32             $data_transaksi[] = array('id' => $value['No'],
33             'id_nasabah' => $value['No'], 'total_pinjaman' =>
34             $value['Total_Pinjaman'], 'agunan' => $value['Tipe_Agunan'],
35             'pendapatan_utama' => $value['Pendapatan_Utama'], 'total_biaya_hidup'
=>
36             $value['Total_Biaya_Hidup'], 'status_pinjaman' =>
37             $value['Status_Pinjaman'] );
38
39             $this->M_Nasabah->input($data_nasabah);
40             $this->M_Transaksi->input($data_transaksi);
41         }
42     }
43     $this->export_csv();
44     $this->get_data_from_data_train();
45     $this->convert_datatrain_to_arff();
46     $this->create_model();
47
48     $this->session->set_flashdata("pesan", "<div class=\"alert
49     alert-success\" style=\"pointer:cursor;position: fixed; z-index:
50     9999;top: 0px;margin-top: 50px; margin-left: 10px; width:98.5%;\"><a
51     href=\"#\" class=\"close\" data-dismiss=\"alert\" aria-
52     label=\"close\">&times;</a> Penambahan Dataset Berhasil!!! </div>");
53
54     redirect(base_url('dataset'));

```



```

46 }
47 else {
48     $this->session->set_flashdata("pesan", "<div class=\"alert
49 alert-danger\" style=\"pointer:cursor;position: fixed; z-index:
50 9999;top: 0px;margin-top: 50px; margin-left: 10px; width:98.5%;\"><a
51 href=\"\"#\" class=\"close\" data-dismiss=\"alert\" aria-
52 label=\"close\">&times;</a> Penambahan Dataset Gaga!!! </div>");
53     redirect(base_url('dataset'));
54 }

```

Kode Program 5. 1 Fungsi *Input Dataset*

Kode Program 5.1 menunjukkan kode pemrograman untuk implementasi fungsi *input dataset* yang dimulai dengan pengecekan apakah terdapat *file upload_datatrain.csv* di direktori *assets/uploads/*. Apabila terdapat *file upload_datatrain.csv* maka akan dilakukan penghapusan *file*. Selanjutnya dilakukan pengkonfigurasi terhadap *file dataset* yang akan diunggah nantinya. Konfigurasi yang dilakukan adalah konfigurasi terhadap direktori *upload*, ekstensi *file* yang diperbolehkan, ukuran maksimal *file*, dan juga penamaan *file* setelah diunggah. Tahap selanjutnya adalah proses pengunggahan *file* yang didalamnya terdapat pemanggilan *external library csvimport*. *Library csvimport* akan membaca isi dari *file csv* yang telah diunggah dan dikonversikan ke dalam bentuk *array*. Sebelum dilakukan pemrosesan data dari *file csv*, sebelumnya dilakukan pengecekan terhadap jumlah *row* di *table* nasabah. Apabila terdapat *row* lebih dari nol, maka dilakukan penghapusan data baik di *table* nasabah maupun di *table* transaksi. Hal ini dimaksudkan agar *dataset* yang diunggah tidak mengalami redundansi data nantinya. Setelah tahap pengecekan *row* di *table* nasabah selesai, dilakukan pemrosesan data dari *file csv* yang telah diunggah untuk kemudian disimpan di *table* nasabah dan *table* transaksi di *database*.

Selanjutnya dilakukan pemanggilan fungsi *export_csv()* guna mengonversikan data dengan *field* terpilih dari *database* untuk dijadikan *file csv* dan sekaligus memasukkan data ke *table datatrain*. Data yang tersimpan di *table datatrain* nantinya akan bertindak sebagai *data train* untuk sistem yang akan dibangun. Setelah itu dilakukan pemanggilan fungsi *get_data_from_data_train()*. Fungsi *get_data_from_data_train()* berfungsi untuk mendapatkan beberapa data dari *table datatrain* untuk nantinya disimpan ke *table datatest*. Tujuan dari pengambilan beberapa data dari *table datatrain* agar pembentukan *header* untuk pembuatan *file arff* nantinya dapat sesuai antara *file* yang dijadikan sebagai *data train* dengan data yang dijadikan sebagai *data test*. Kemudian, dilakukan pemanggilan fungsi *convert_data_train_to_arff()* untuk mengonversikan *file datatrain* berekstensi *csv* menjadi *file* berekstensi *arff*. Terakhir, dilakukan pemanggilan fungsi *create_model()* untuk mengonversikan *file arff* hasil dari pemrosesan fungsi *convert_data_train_to_arff()* menjadi *file* model. *File* model hasil konversi mempunyai fungsi yang sama dengan pemanggilan fungsi C4.5 yang telah disediakan oleh *weka*.

```

public function export_csv()
1  $filename = 'datatrain.csv';
2  $records = $this->M_Nasabah->get_records_to_create_csv();
3
4  $data = "JENIS_KELAMIN, TOTAL_PINJAMAN, JUMLAH_TANGGUNGAN,
5  STATUS_PERNIKAHAN, PENDAPATAN_UTAMA, TOTAL_BIAYA_HIDUP,
6  STATUS_PINJAMAN \n"; //Column headers
7  foreach ($records as $key => $value) {
8      $jenis_kelamin = NULL;
9      $total_pinjaman = NULL;
10     $jumlah_tanggungan = NULL;
11     $pendapatan_utama = NULL;
12     $total_biaya_hidup = NULL;
13     $status_pernikahan = NULL;
14     $id = $key + 1;
15
16     if ($value->jenis_kelamin == "LAKI-LAKI") {
17         $jenis_kelamin = "L";
18     }
19     elseif ($value->jenis_kelamin == "PEREMPUAN") {
20         $jenis_kelamin = "P";
21     }
22
23     // transformation data jumlah tanggungan
24     if ($value->jumlah_tanggungan >= 0 && $value->jumlah_tanggungan <= 2) {
25         $jumlah_tanggungan = "SEDIKIT";
26     }
27     elseif ($value->jumlah_tanggungan >= 3 && $value->jumlah_tanggungan <= 4) {
28         $jumlah_tanggungan = "SEDANG";
29     }
30     else{
31         $jumlah_tanggungan = "BANYAK";
32     }
33
34     // transformation data status pernikahan
35     if ($value->status_pernikahan == "MENIKAH") {
36         $status_pernikahan = "MENIKAH";
37     }
38     elseif ($value->status_pernikahan == "BELUM MENIKAH") {
39         $status_pernikahan = "BELUM_MENIKAH";
40

```



```

41     }
42     elseif ($value->status_pernikahan == "CERAI") {
43         $status_pernikahan = "CERAI";
44     }
45
46     // transformation data total pinjaman
47     if ($value->total_pinjaman >= 0 && $value->total_pinjaman <=
48 2500000) {
49         $total_pinjaman = "SANGAT_RENDAH";
50     }
51     elseif ($value->total_pinjaman >= 2500001 && $value-
52 >total_pinjaman <= 4500000) {
53         $total_pinjaman = "RENDAH";
54     }
55     elseif ($value->total_pinjaman >= 4500001 && $value-
56 >total_pinjaman <= 6000000) {
57         $total_pinjaman = "SEDANG";
58     }
59     elseif ($value->total_pinjaman >= 6000001 && $value-
60 >total_pinjaman <= 8000000) {
61         $total_pinjaman = "TINGGI";
62     }
63     else{
64         $total_pinjaman = "SANGAT_TINGGI";
65     }
66
67     // transformation data pendapatan utama
68     if ($value->pendapatan_utama >= 0 && $value->pendapatan_utama
69 <= 1500000) {
70         $pendapatan_utama = "RENDAH";
71     }
72     elseif ($value->pendapatan_utama >= 1500001 && $value-
73 >pendapatan_utama <= 3000000) {
74         $pendapatan_utama = "SEDANG";
75     }
76     else{
77         $pendapatan_utama = "TINGGI";
78     }
79
80     if ($value->total_biaya_hidup >= 0 && $value-
81 >total_biaya_hidup <= 1500000) {
82         $total_biaya_hidup = "RENDAH";
83     }
84     elseif ($value->total_biaya_hidup >= 1500001 && $value-

```

```

82 >total_biaya_hidup <= 3000000) {
83     $total_biaya_hidup = "SEDANG";
84 }
85 else{
86     $total_biaya_hidup = "TINGGI";
87 }
88
89 $data.=
90 $jenis_kelamin.','.$total_pinjaman.','.$jumlah_tanggungan.','.$stat
91 us_pernikahan.','.$pendapatan_utama.','.$total_biaya_hidup.','.$val
92 ue->status_pinjaman."\n"; //Append data to csv
93
94 $data_train[] = array('id' => $id, 'jenis_kelamin' =>
95 $jenis_kelamin, 'total_pinjaman' => $total_pinjaman,
96 'jumlah_tanggungan' => $jumlah_tanggungan, 'status_pernikahan' =>
97 $status_pernikahan, 'pendapatan_utama' => $pendapatan_utama,
98 'total_biaya_hidup' => $total_biaya_hidup, 'status_pinjaman' =>
99 $value->status_pinjaman );
100 }
101 $check_num = $this->M_Datatrain->get_num_rows_data();
102 if ($check_num > 0) {
103     $this->M_Datatrain->delete_all_data();
104 }
105 $this->M_Datatrain->input($data_train);
106 if (file_exists("./assets/uploads/datatrain.csv")) {
107     unlink("./assets/uploads/datatrain.csv");
108 }
109 file_put_contents('./assets/uploads/' . $filename, $data);

```

Kode Program 5. 2 Fungsi *export_csv()*

Kode Program 5.2 menunjukkan kode pemrograman untuk implementasi fungsi *export_csv()* yang dimulai dengan pemberian nama untuk *file csv* yang akan terbentuk nantinya. Kemudian, dilakukan pemanggilan fungsi ke model untuk memperoleh data dari *table* nasabah dan transaksi dalam bentuk *object*. Setelah itu, dilakukan pembuatan *header* untuk *file csv* yang akan dikonversi nantinya. Karena, data mentah hasil dari proses pengunggahan *file csv* ke *table* nasabah dan transaksi masih dalam bentuk data mentah, maka perlu dilakukan proses transformasi data menjadi bentuk *range value* seperti yang telah dilakukan di proses transformasi data pada Bab Perancangan. Data hasil transformasi data kemudian akan digabungkan dengan *header* yang telah terbentuk sebelumnya.

Sebelum data dimasukkan ke *table datatrain*, dilakukan pengecekan terlebih dahulu apakah terdapat *row* di *table datatrain*. Apabila terdapat *row* lebih dari nol maka dilakukan penghapusan keseluruhan data di *table datatrain*. Setelah itu dilakukan pemasukan data ke *table datatrain*. Untuk menghindari *file* redundansi, maka dilakukan pengecekan apakah terdapat *file* dengan nama *datatrain.csv* di direktori */assets/uploads*, apabila terdapat *file* dengan nama

datatrain.csv akan dilakukan penghapusan *file*. Tahap terakhir adalah penempatan *file csv* yang telah diproses ke direktori yang telah ditentukan.

public function get_data_from_data_train()	
1	\$check_num = \$this->M_Datatest->get_num_rows_data();
2	if (\$check_num > 0) {
3	\$this->M_Datatest->delete_all_data();
4	}
5	
6	\$data_jenis_kelamin = \$this->M_Datatrain->get_jenis_kelamin();
7	\$count_jenis_kelamin = count(\$data_jenis_kelamin);
8	foreach (\$data_jenis_kelamin as \$key => \$jk) {
9	\$jjenis_kelamin[] = \$jk->jenis_kelamin;
10	}
11	
12	\$data_total_pinjaman = \$this->M_Datatrain->get_total_pinjaman();
13	\$count_total_pinjaman = count(\$data_total_pinjaman);
14	foreach (\$data_total_pinjaman as \$key => \$tp) {
15	\$total_pinjaman[] = \$tp->total_pinjaman;
16	}
17	
18	\$data_jumlah_tanggungan = \$this->M_Datatrain->
19	get_jumlah_tanggungan();
20	\$count_jumlah_tanggungan = count(\$data_jumlah_tanggungan);
21	foreach (\$data_jumlah_tanggungan as \$key => \$jt) {
22	\$jjumlah_tanggungan[] = \$jt->jumlah_tanggungan;
23	}
24	
25	\$data_status_pernikahan = \$this->M_Datatrain->
26	get_status_pernikahan();
27	\$count_status_pernikahan = count(\$data_status_pernikahan);
28	foreach (\$data_status_pernikahan as \$key => \$sp) {
29	\$status_pernikahan[] = \$sp->status_pernikahan;
30	}
31	
32	\$data_pendapatan_utama = \$this->M_Datatrain->get_pendapatan_utama();
33	\$count_pendapatan_utama = count(\$data_pendapatan_utama);
34	foreach (\$data_pendapatan_utama as \$key => \$dp) {
35	\$pendapatan_utama[] = \$dp->pendapatan_utama;
36	}
	\$data_total_biaya_hidup = \$this->M_Datatrain->

```

37 >get_total_biaya_hidup();
38 $count_total_biaya_hidup = count($data_total_biaya_hidup);
39 foreach ($data_total_biaya_hidup as $key => $tbh) {
40     $total_biaya_hidup[] = $tbh->total_biaya_hidup;
41 }
42
43 $data_status_pinjaman = $this->M_Datatrain->get_status_pinjaman();
44 $count_status_pinjaman = count($data_status_pinjaman);
45 foreach ($data_status_pinjaman as $key => $spj) {
46     $status_pinjaman[] = $spj->status_pinjaman;
47 }
48
49 for ($i=1; $i <= 30; $i++) {
50     $jk_rand = $jenis_kelamin[rand(0, $count_jenis_kelamin-1)];
50     $tp_rand = $total_pinjaman[rand(0, $count_total_pinjaman-1)];
51     $jt_rand = $jumlah_tanggungan[rand(0, $count_jumlah_tanggungan-1)];
52     $sp_rand = $status_pernikahan[rand(0, $count_status_pernikahan-1)];
53     $pu_rand = $pendapatan_utama[rand(0, $count_pendapatan_utama-1)];
54     $tbh_rand = $total_biaya_hidup[rand(0, $count_total_biaya_hidup-1)];
55     $spj_rand = $status_pinjaman[rand(0, $count_status_pinjaman-1)];
56     $data[] = array('id' => $i, 'jenis_kelamin' => $jk_rand,
57 'total_pinjaman' => $tp_rand, 'jumlah_tanggungan' => $jt_rand,
58 'status_pernikahan' => $sp_rand, 'pendapatan_utama' => $pu_rand,
59 'total_biaya_hidup' => $tbh_rand, 'status_pinjaman' => $spj_rand);
60 }
61 $this->M_Datatest->insert_batch($data);
62

```

Kode Program 5. 3 Fungsi *get_data_from_data_train()*

Kode Program 5.3 menunjukkan implementasi untuk fungsi *get_data_from_data_train()* yang dimulai dengan pengecekan row untuk *table datatest*. Apabila row lebih dari nol maka akan dilakukan penghapusan data pada *table datatest*. Selanjutnya dilakukan pengambilan data dalam bentuk *object* dan jumlah data dari *table datatrain* untuk *field* jenis kelamin, total pinjaman, jumlah tanggungan, status pernikahan, pendapatan utama, total biaya hidup, dan status pinjaman. Kemudian dilakuan perulangan sebanyak 30 kali dengan memanfaatkan fungsi *rand* agar data yang tersaji menjadi acak. Hasil pengacakan data kemudian akan dimasukkan ke *table datatest*.

```

public function convert_datatrain_to_arff()

```

```

1 if (file_exists("./assets/uploads/datatrain.arff")) {
2     unlink("./assets/uploads/datatrain.arff");

```

3	}
4	
5	\$cmd = "java -cp C:\xampp\htdocs\KSP-MitraRaya\assets\weka.jar
6	weka.core.converters.CSVLoader C:\xampp\htdocs\KSP-
7	MitraRaya\assets\uploads\datatrain.csv > C:\xampp\htdocs\KSP-
8	MitraRaya\assets\uploads\datatrain.arff";
9	exec(\$cmd, \$output);

Kode Program 5. 4 Fungsi *convert_datatrain_to_arff()*

Kode Program 5.4 menunjukkan implementasi fungsi *convert_datatrain_to_arff()* yang dimulai dengan pengecekan apakah terdapat file *datatrain.arff* di direktori */assets/uploads/*. Apabila terdapat file *datatrain.arff* akan dilakukan penghapusan file *datatrain.arff*. Kemudian dilakukan pengeksekusian program *weka* dengan memanfaatkan fungsi *exec* untuk mengkonversi *datatrain* berektensi *csv* menjadi format *arff* agar bisa dibaca oleh *weka* nantinya.

public function create_model()	
1	if (file_exists("../assets/uploads/datatrain.model")) {
2	unlink("../assets/uploads/datatrain.model");
3	}
4	\$cmd = shell_exec("java -cp C:\xampp\htdocs\KSP-
5	MitraRaya\assets\weka.jar weka.classifiers.trees.J48 -t
6	C:\xampp\htdocs\KSP-MitraRaya\assets\uploads\datatrain.arff -d
	C:\xampp\htdocs\KSP-MitraRaya\assets\uploads\datatrain.model");

Kode Program 5. 5 Fungsi *create_model()*

Kode Program 5.5 menunjukkan implementasi fungsi *create_model()* yang diawali dengan pengecekan apakah terdapat file *datatrain.model* di direktori */assets/uploads/*. Apabila terdapat file *datatrain.model* akan dilakukan penghapusan terhadap file *datatrain.model*. Selanjutnya dilakukan proses pengkonversian dari *data train* menjadi *file model* dengan menggunakan fungsi dari *weka*. Eksekusi fungsi *weka* menggunakan fungsi *shell_exec*.

5.3.2 Implementasi Fungsi Prediksi Nasabah

Fungsi prediksi nasabah terdapat pada kelas *Controller* *PrediksiNasabah*, yang mana merupakan fungsi yang dapat digunakan oleh Pimpinan KSP untuk memprediksi pengajuan nasabah baru.

public function proses_submit()	
1	\$nama = \$this->input->post('nama');
2	\$jenis_kelamin = \$this->input->post('jenis_kelamin');
3	\$total_pinjaman = \$this->input->post('total_pinjaman');
4	\$jumlah_tanggungan = \$this->input->post('jumlah_tanggungan');

```

5  $status_perkawinan = $this->input->post('status_perkawinan');
6  $pendapatan_utama = $this->input->post('pendapatan_utama');
7  $total_biaya_hidup = $this->input->post('total_biaya_hidup');
8  $status_pinjaman = "MACET";
9  $ttotal_pinjaman = NULL;
10 $tjumlah_tanggungan = NULL;
11 $tpendapatan_utama = NULL;
12 $ttotal_biaya_hidup = NULL;
13
14 // transformation data total pinjaman
15 if ($total_pinjaman >= 0 && $total_pinjaman <= 2500000) {
16     $ttotal_pinjaman = "SANGAT_RENDAH";
17 }
18 elseif ($total_pinjaman >= 2500001 && $total_pinjaman <= 4500000) {
19     $ttotal_pinjaman = "RENDAH";
20 }
21 elseif ($total_pinjaman >= 4500001 && $total_pinjaman <= 6000000) {
22     $ttotal_pinjaman = "SEDANG";
23 }
24 elseif ($total_pinjaman >= 6000001 && $total_pinjaman <= 8000000) {
25     $ttotal_pinjaman = "TINGGI";
26 }
27 else{
28     $ttotal_pinjaman = "SANGAT_TINGGI";
29 }
30
31 // transformation data jumlah tanggungan
32 if ($jumlah_tanggungan >= 0 && $jumlah_tanggungan <= 2) {
33     $tjumlah_tanggungan = "SEDIKIT";
34 }
35 elseif ($jumlah_tanggungan >= 3 && $jumlah_tanggungan <= 4) {
36     $tjumlah_tanggungan = "SEDANG";
37 }
38 else{
39     $tjumlah_tanggungan = "BANYAK";
40 }
41
42 // transformation data pendapatan utama
43 if ($pendapatan_utama >= 0 && $pendapatan_utama <= 1500000) {
44     $tpendapatan_utama = "RENDAH";
45 }

```



```

46 elseif ($pendapatan_utama >= 1500001 && $pendapatan_utama <=
47 3000000) {
48     $tpendapatan_utama = "SEDANG";
49 }
50 else{
51     $tpendapatan_utama = "TINGGI";
52 }
53 // transformation data total biaya hidup
54 if ($total_biaya_hidup >= 0 && $total_biaya_hidup <= 1500000) {
55     $ttotal_biaya_hidup = "RENDAH";
56 }
57 elseif ($total_biaya_hidup >= 1500001 && $total_biaya_hidup <=
58 3000000) {
59     $ttotal_biaya_hidup = "SEDANG";
60 }
61 else{
62     $ttotal_biaya_hidup = "TINGGI";
63 }
64 $id = NULL;
65 $id_pengajuan = NULL;
66 $last_records = $this->M_Datatest->get_last_record();
67 foreach ($last_records as $key => $value) {
68     $id = $value->id+1;
69 }
70
71 $last_record_pengajuan = $this->M_Pengajuan->get_data();
72 if (count($last_record_pengajuan) == 0) {
73     $id_pengajuan = 1;
74 }
75 else{
76     foreach ($last_record_pengajuan as $key => $last) {
77         $id_pengajuan = $last->id + 1;
78     }
79 }
80
81 $date = date('Y-m-d');
82
83 $data = array('id' => $id, 'jenis_kelamin' => $jenis_kelamin,
84 'total_pinjaman' => $ttotal_pinjaman, 'jumlah_tanggungan' =>
85 $tjumlah_tanggungan, 'status_pernikahan' => $status_perkawinan,
86 'pendapatan_utama' => $tpendapatan_utama, 'total_biaya_hidup' =>

```

```

86 $ttotal_biaya_hidup, 'status_pinjaman' => $status_pinjaman);
87 $data_pengajuan = array('id' => $id_pengajuan, 'nama' => $nama,
88 'jenis_kelamin' => $jenis_kelamin, 'total_pinjaman' =>
89 $total_pinjaman, 'jumlah_tanggungan' => $jumlah_tanggungan,
90 'status_pernikahan' => $status_perkawinan, 'pendapatan_utama' =>
91 $pendapatan_utama, 'total_biaya_hidup' => $total_biaya_hidup,
92 'tgl_pengajuan' => $date);
93
94 $this->M_Datatest->input($data);
95 $this->M_Pengajuan->input($data_pengajuan);
96 $this->export_csv_predict();
97 $this->convert_datatest_to_arff();
98 $this->make_predict($id_pengajuan);
99
100 $data['pengajuan'] = $this->M_Pengajuan->get_last_record();
101 $this->load->view('prediksi_nasabah/hasil_prediksi.php', $data);

```

Kode Program 5. 6 Fungsi Prediksi Nasabah

Kode Program 5.6 menunjukkan implementasi fungsi prediksi nasabah yang diawali dengan pendefinisian beberapa *variable* dan diinisiasi dengan nilai yang didapatkan dari masukan *form user* maupun inisiasi secara manual. Setelah itu dilakukan proses transformasi data terhadap beberapa variabel diantaranya adalah variabel total pinjaman, jumlah tanggungan, pendapatan utama, dan total biaya hidup menjadi *value range* sesuai dengan proses transformasi data pada Bab Perancangan. Setelah itu dilakukan pemasukan data yang telah diproses ke *table datatest* dan data mentah sebelum dilakukan transformasi data ke *table pengajuan*.

Kemudian, dilakukan pemanggilan fungsi *export_csv_predict()* untuk mengonversikan data yang telah tersimpan di *table datatest* menjadi bentuk *file* berektensi *csv*. *File* yang telah terkonversi dalam *format csv* akan dikonversi lagi dalam *format arff* dengan memanggil fungsi *convert_datatest_to_arff()*. Tahap terakhir untuk dapat memprediksi pengajuan nasabah adalah dengan memanggil fungsi *make_predict()*. Hasil prediksi nasabah akan ditampilkan ke *view*.

```

public function export_csv_predict()
1   if (file_exists("./assets/uploads/datatest.csv")) {
2       unlink("./assets/uploads/datatest.csv");
3   }
4   $filename = 'datatest.csv';
5   $records = $this->M_Datatest->get_data_desc();
6   $data =
7   "JENIS_KELAMIN,TOTAL_PINJAMAN,JUMLAH_TANGGUNGAN,STATUS_PERNIKAHAN,PE
NDAPATAN_UTAMA,TOTAL_BIAYA_HIDUP,STATUS_PINJAMAN\n"; //Column
8   headers

```

```

9  foreach ($records as $key => $value) {
10     $data.=                               $value->jenis_kelamin.','.$value-
11     >total_pinjaman.','.$value->jumlah_tanggungan.','.$value-
12     >status_pernikahan.','.$value->pendapatan_utama.','.$value-
13     >total_biaya_hidup.','.$value->status_pinjaman."\n"; //Append data
14     to csv
15 }
16 file_put_contents('./assets/uploads/' . $filename, $data);

```

Kode Program 5. 7 Fungsi *export_csv_predict()*

Kode Program 5.7 menunjukkan implementasi fungsi *export_csv_predict()* yang dimulai dengan pengecekan apakah terdapat *file datatest.csv* pada direktori */assets/uploads/*. Apabila terdapat *file* tersebut akan dilakukan penghapusan terhadap *file datatest.csv*. Setelah itu dilakukan pengambilan data dari *table datatest* dalam bentuk *object*. Kemudian, dibuat *header* untuk *file csv* yang akan dibuat dan diikuti dengan penambahan data yang telah didapat sebelumnya dalam bentuk *object* ke *header* yang telah dibuat. Tahap terakhir adalah menempatkan *file csv* yang telah terbentuk ke direktori yang telah ditentukan dengan nama yang telah ditentukan juga sebelumnya.

```

public function convert_datatest_to_arff()
1  if (file_exists("./assets/uploads/datatest.arff")) {
2      unlink("./assets/uploads/datatest.arff");
3  }
4  $cmd = "java -cp C:\\xampp\\htdocs\\KSP-MitraRaya\\assets\\weka.jar
5  weka.core.converters.CSVLoader C:\\xampp\\htdocs\\KSP-
6  MitraRaya\\assets\\uploads\\datatest.csv > C:\\xampp\\htdocs\\KSP-
7  MitraRaya\\assets\\uploads\\datatest.arff";
8  exec($cmd, $output);

```

Kode Program 5. 8 Fungsi *convert_datatest_to_arff()*

Kode Program 5.8 menunjukkan implementasi fungsi *convert_datatest_to_arff()* yang diawali dengan pengecekan apakah terdapat *file datatest.arff* pada direktori */assets/uploads*. Apabila terdapat *file datatest.arff* akan dilakukan penghapusan terhadap *file datatest.arff*. Kemudian dilakukan pengonversian dari *file datatest* berekstensi *csv* menjadi ekstensi *arff* dengan menggunakan fungsi *weka* yang dieksekusi lewat fungsi *exec*.

```

public function make_predict($id_pengajuan)
1  $cmd = "java -cp C:\\xampp\\htdocs\\KSP-MitraRaya\\assets\\weka.jar
2  weka.classifiers.misc.InputMappedClassifier -I -trim -L
3  C:\\xampp\\htdocs\\KSP-MitraRaya\\assets\\uploads\\datatrain.model -t
4  C:\\xampp\\htdocs\\KSP-MitraRaya\\assets\\uploads\\datatrain.arff -T
5  C:\\xampp\\htdocs\\KSP-MitraRaya\\assets\\uploads\\datatest.arff -p 0";
6  exec($cmd, $output);
7  $explode = explode(":", $output[17]);
8  $explode2 = explode(" ", $explode[2]);

```

8	<code>\$result = \$explode2[0];</code>
9	<code>\$data_array = array('status_pinjaman' => \$result);</code>
10	<code>\$cond = array('id' => \$id_pengajuan);</code>
	<code>\$this->M_Pengajuan->update(\$data_array, \$cond);</code>

Kode Program 5. 9 Fungsi *make_predict(\$id_pengajuan)*

Kode Program 5.9 menunjukkan implementasi fungsi *make_predict(\$id_pengajuan)* yang diawali dengan proses prediksi dengan memanfaatkan fitur dari *weka* yang dieksekusi dengan fungsi *exec*. Hasil dari proses prediksi dengan menggunakan fungsi *exec* tersebut akan menghasilkan *output* dalam bentuk *array*. *Array* yang dihasilkan tersebut akan diproses lebih lanjut dengan menggunakan fungsi *explode*. Tahap terakhir adalah dengan *update table* pengajuan dengan memasukkan data yang dihasilkan dari proses *explode* yang telah dilakukan.

5.3.3 Implementasi Fungsi Lihat Akurasi

Fungsi lihat akurasi terdapat pada kelas *Controller* *PrediksiNasabah*, yang mana merupakan fungsi yang dapat digunakan oleh Pimpinan KSP untuk melihat tingkat akurasi yang dihasilkan sistem.

	<code>public function lihat_akurasi()</code>
1	<code>\$cmd = shell_exec("java -cp C:/xampp/htdocs/KSP-MitraRaya/assets/weka.jar weka.classifiers.trees.J48 -t C:/xampp/htdocs/KSP-MitraRaya/assets/uploads/datatrain.arff -v -o");</code>
2	
3	
4	<code>\$data['row'] = \$cmd;</code>
5	<code>\$this->load->view('prediksi_nasabah/lihat_akurasi.php', \$data);</code>

Kode Program 5. 10 Fungsi Lihat Akurasi

Kode Program 5.10 menunjukkan implementasi fungsi lihat akurasi yang diawali dengan mengeksekusi fitur *weka* untuk melihat akurasi yang dihasilkan berdasarkan algoritme C4.5 dengan menggunakan fungsi *shell_exec*. Luaran dari fungsi tersebut berupa *String* yang akan ditampilkan ke *view*.

5.3.4 Implementasi Fungsi Lihat Tree

	<code>public function lihat_tree()</code>
1	<code>\$cmd = shell_exec("java -cp C:/xampp/htdocs/KSP-MitraRaya/assets/weka.jar weka.classifiers.trees.J48 -t C:/xampp/htdocs/KSP-MitraRaya/assets/uploads/datatrain.arff -v -no-cv");</code>
2	
3	
4	<code>\$data['row'] = \$cmd;</code>
5	
6	<code>\$this->load->view('prediksi_nasabah/lihat_tree.php', \$data);</code>

Kode Program 5. 11 Fungsi Lihat Tree

Fungsi lihat *tree* terdapat pada kelas *Controller* *PrediksiNasabah*, yang mana merupakan fungsi yang dapat digunakan oleh Pimpinan KSP untuk melihat

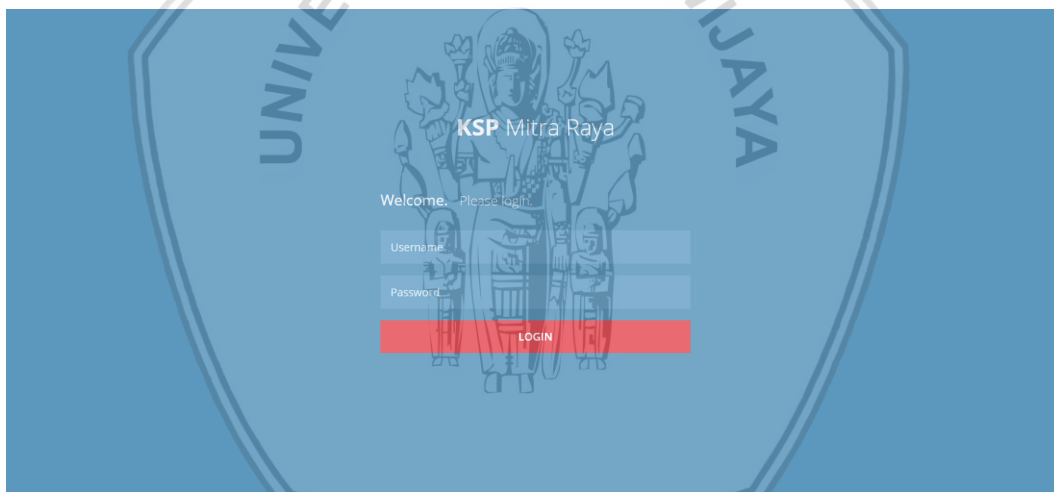
tingkat *tree* yang dihasilkan sistem. Fungsi ini diimplementasikan dalam Kode Program 5.11 yang diawali dengan mengeksekusi fitur *weka* untuk membangun *tree* yang dihasilkan berdasarkan algoritme C4.5 dengan menggunakan fungsi *shell_exec*. Luaran dari fungsi tersebut berupa *String* yang akan ditampilkan ke *view*.

5.4 Implementasi Visualisasi

Implementasi visualisasi merupakan tampilan dari sistem yang akan dibangun. Proses pembuatan tampilan dilakukan dengan memanfaatkan HTML, CSS, dan *JavaScript*. Terdapat tujuh tampilan visualisasi yaitu halaman *login*, *dashboard*, prediksi nasabah, lihat akurasi, lihat *tree*, *input dataset*, dan data nasabah.

5.4.1 Implementasi Visualisasi Halaman *Login*

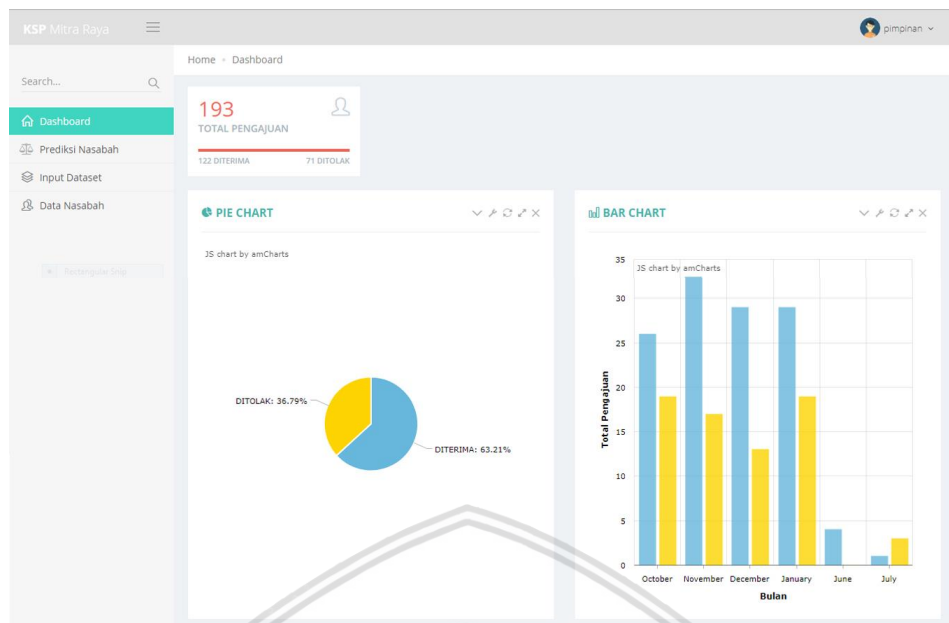
Gambar 5.1 merupakan penggambaran hasil implementasi visualisasi halaman *login*. Halaman *login* merupakan halaman pertama yang akan ditampilkan dan berfungsi untuk memasukkan *username* dan *password* untuk bisa masuk ke sistem.



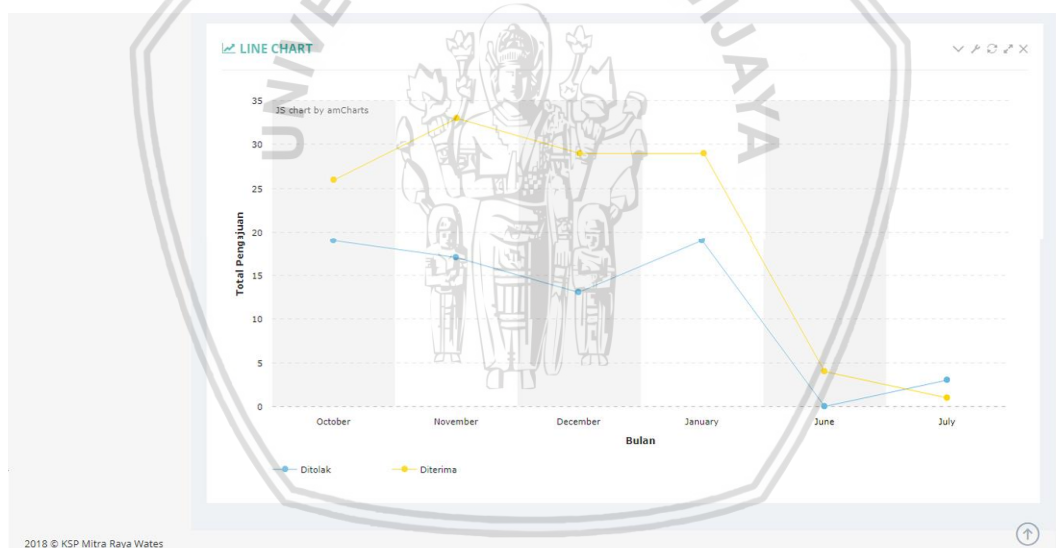
Gambar 5. 1 Implementasi Visualisasi Halaman *Login*

5.4.2 Implementasi Visualisasi Halaman *Dashboard*

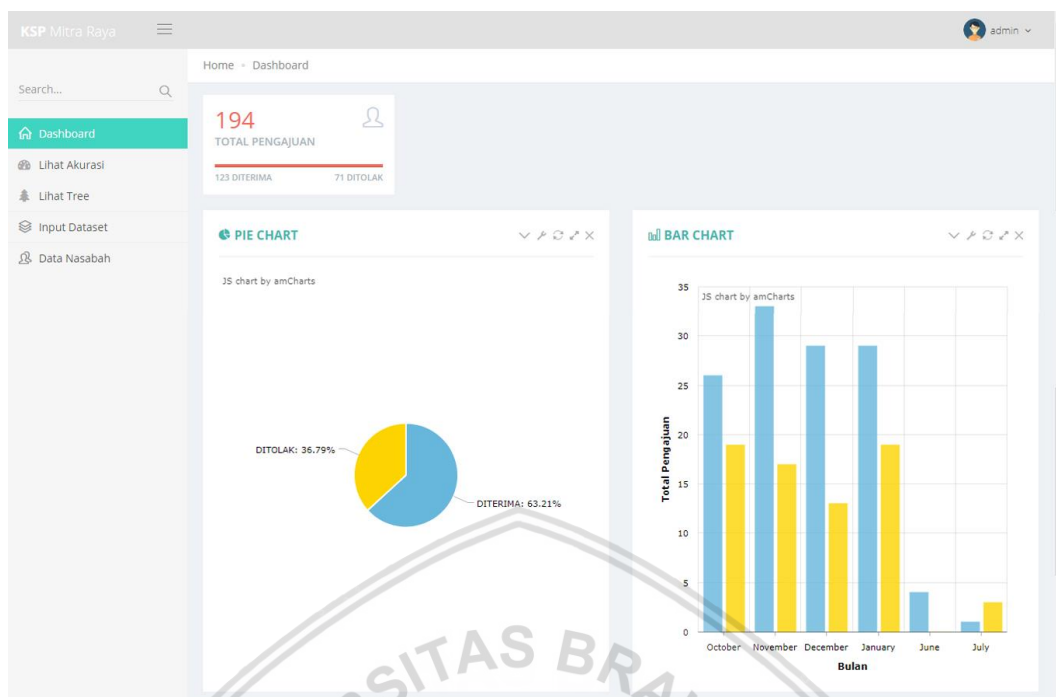
Gambar 5.2 dan 5.3 merupakan penggambaran hasil implementasi visualisasi halaman *dashboard* untuk aktor Pimpinan KSP dan Gambar 5.4 dan 5.5 merupakan penggambaran hasil implementasi visualisasi halaman *dashboard* untuk aktor Admin KSP. Pada halaman *dashboard* akan ditampilkan total pengajuan berikut dengan total pengajuan yang diterima maupun ditolak. Selain itu juga terdapat persentase total pengajuan diterima maupun ditolak dalam bentuk *pie chart*, *time-series* pengajuan yang ditampilkan dalam bentuk *bar chart*, dan juga *trend* pengajuan tiap bulan yang ditampilkan dalam bentuk *line chart*.



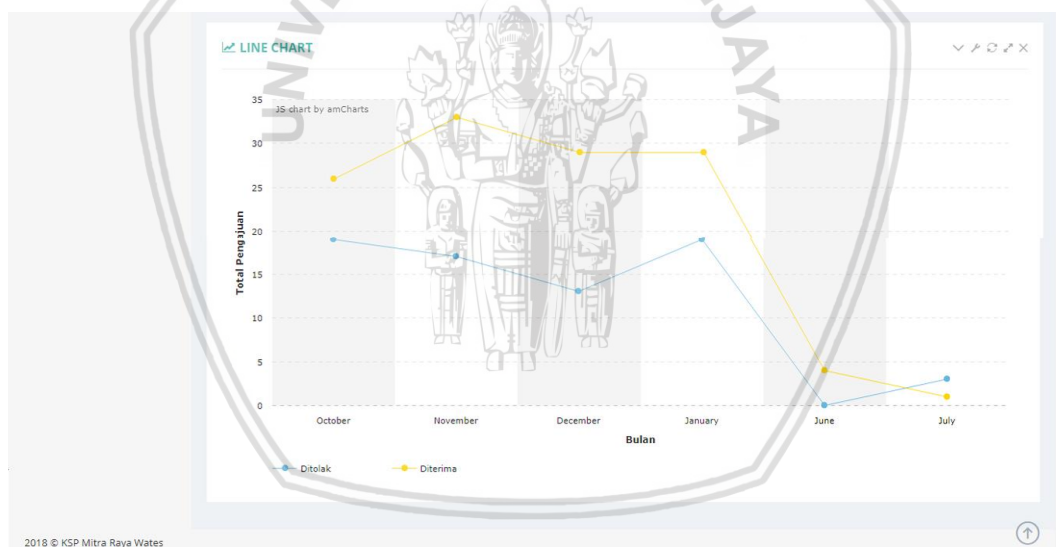
Gambar 5. 2 Implementasi Visualisasi Halaman *Dashboard* Pimpinan KSP



Gambar 5. 3 Implementasi Visualisasi Halaman *Dashboard* Pimpinan KSP (lanjutan)



Gambar 5. 4 Implementasi Visualisasi Halaman *Dashboard* Admin KSP



Gambar 5. 5 Implementasi Visualisasi Halaman *Dashboard* Admin KSP (lanjutan)

5.4.3 Implementasi Visualisasi Halaman Prediksi Nasabah

Gambar 5.6 merupakan hasil implementasi visualisasi halaman prediksi nasabah. Halaman ini berfungsi untuk memasukkan data pengajuan nasabah baru dalam bentuk *form*. Hasil dari *submit form* akan menghasilkan tampilan seperti terlihat dalam Gambar 5.7.

KSP Mitra Raya

Home • Prediksi Nasabah • Submit Data

Search...

Dashboard

Prediksi Nasabah

Input Dataset

Data Nasabah

Menu Submit Data dapat digunakan untuk memprediksi pengajuan pinjaman baru berdasarkan parameter Jenis Kelamin, Total Pinjaman, Jumlah Tanggungan, Status Perkawinan, Pendapatan Utama, dan Total Biaya Hidup

FORM SUBMIT

Nama *

Jenis Kelamin * Select

Total Pinjaman *

Jumlah Tanggungan *

Status Perkawinan * Select

Pendapatan Utama *

Total Biaya Hidup *

Submit Reset

2018 © KSP Mitra Raya Wates

Gambar 5. 6 Implementasi Visualisasi Halaman Prediksi Nasabah

KSP Mitra Raya

Home • Prediksi Nasabah • Submit Data • Hasil Prediksi

Search...

Dashboard

Prediksi Nasabah

Input Dataset

Data Nasabah

HASIL PREDIKSI

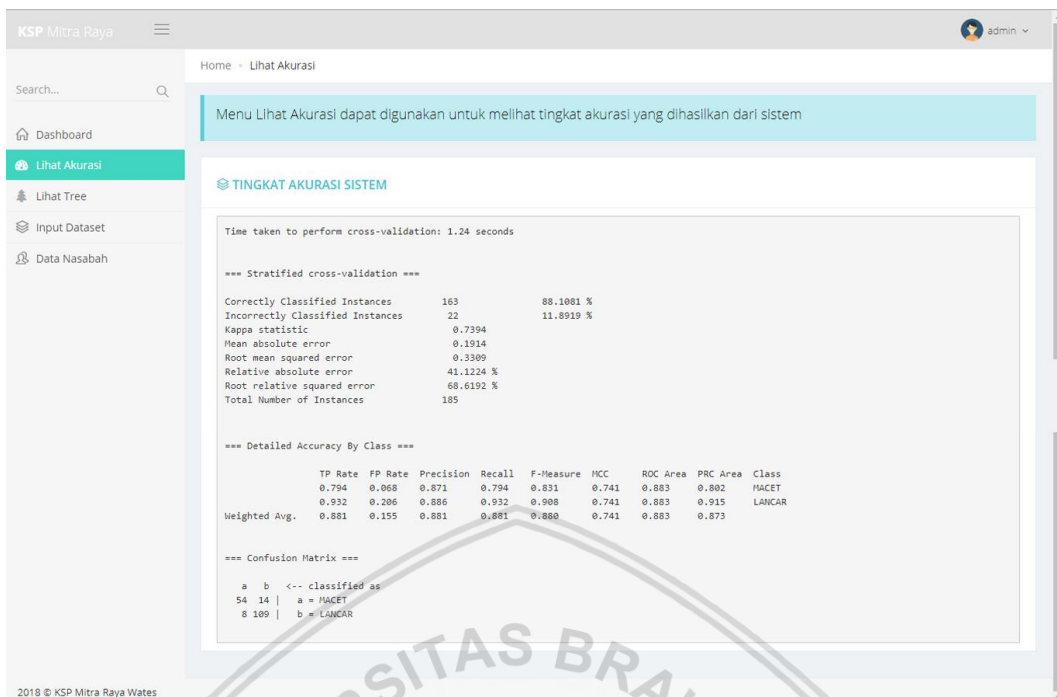
Nama:	SADI
Jenis Kelamin:	L
Total Pinjaman:	5000000
Jumlah Tanggungan:	2
Status Pernikahan:	BELUM_MENIKAH
Pendapatan Utama:	2500000
Total Biaya Hidup:	1500000
Hasil Prediksi:	LANCAR

2018 © KSP Mitra Raya Wates

Gambar 5. 7 Implementasi Visualisasi Halaman Hasil Prediksi Nasabah

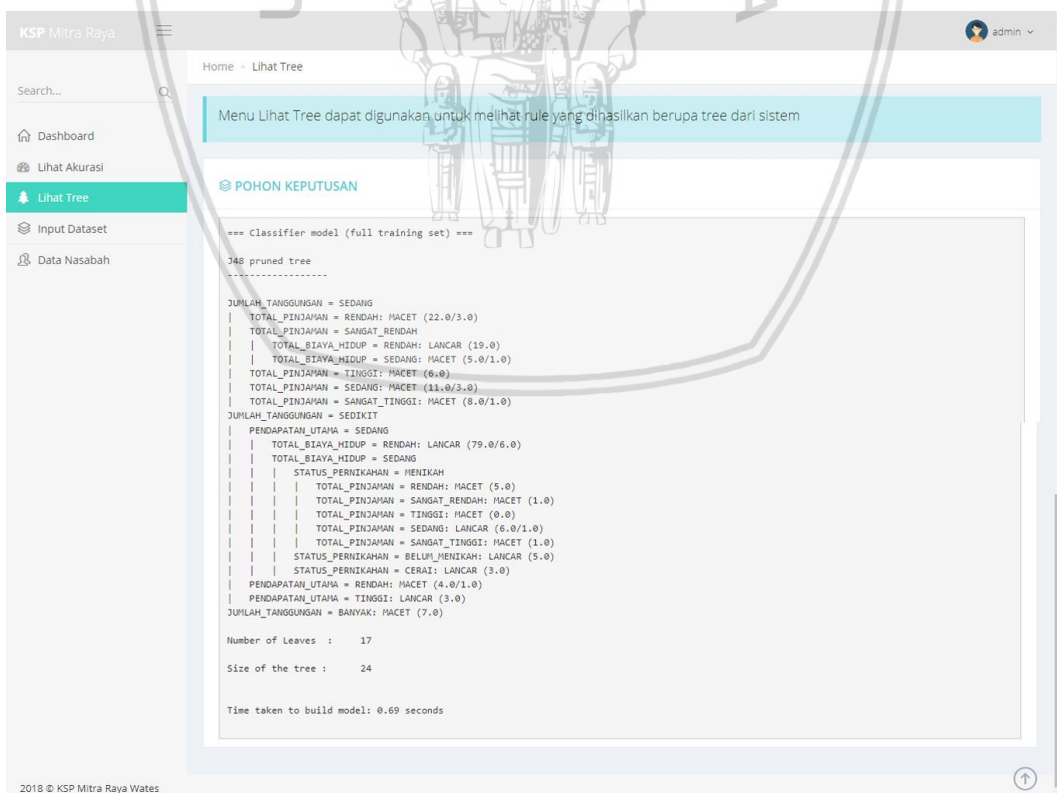
5.4.4 Implementasi Visualisasi Halaman Lihat Akurasi

Gambar 5.8 merupakan hasil implementasi visualisasi halaman lihat akurasi. Akurasi yang didapatkan berasal dari hasil pemrosesan dengan menggunakan *tool weka* menggunakan algoritme C4.5.



Gambar 5. 8 Implementasi Visualisasi Halaman Lihat Akurasi

5.4.5 Implementasi Visualisasi Halaman Lihat Tree



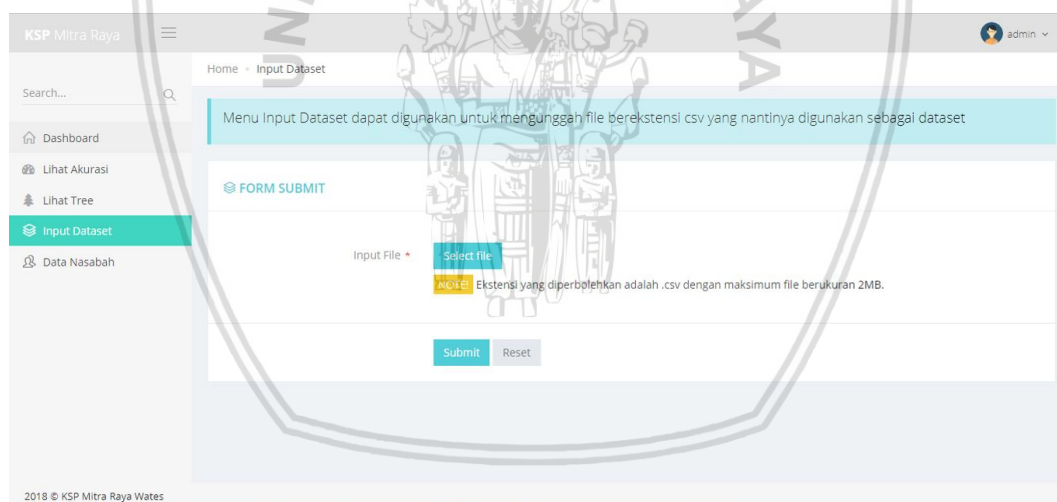
Gambar 5. 9 Implementasi Visualisasi Halaman Lihat Tree

Gambar 5.9 merupakan hasil implementasi visualisasi halaman lihat *tree*. *Tree* yang didapatkan berasal dari hasil pemrosesan dengan menggunakan *tool weka* menggunakan algoritme C4.5.

5.4.6 Implementasi Visualisasi Halaman *Input Dataset*



Gambar 5. 10 Implementasi Visualisasi Halaman *Input Dataset* Pimpinan KSP



Gambar 5. 11 Implementasi Visualisasi Halaman *Input Dataset* Admin KSP

Gambar 5.10 merupakan hasil implementasi visualisasi halaman *input dataset* untuk aktor Pimpinan KSP dan Gambar 5.11 merupakan hasil implementasi visualisasi halaman *input dataset* untuk aktor Admin KSP. Halaman ini dapat digunakan oleh Pimpinan KSP/Admin KSP untuk mengunggah *file* berekstensi *csv* agar nantinya dapat digunakan sebagai *data training* untuk melakukan prediksi nasabah.

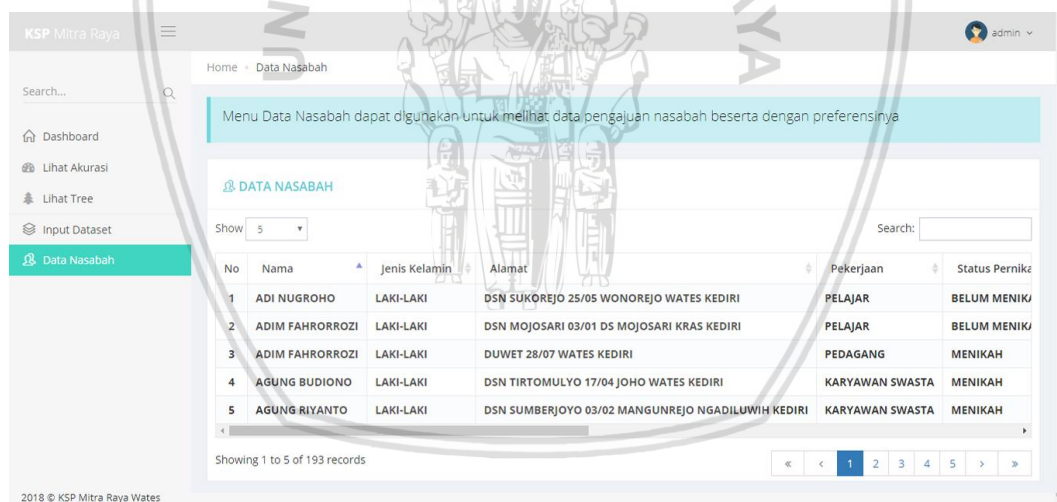
5.4.7 Implementasi Visualisasi Halaman Data Nasabah

Gambar 5.12 merupakan hasil implementasi visualisasi halaman data nasabah untuk aktor Pimpinan KSP dan Gambar 5.13 merupakan hasil implementasi

visualisasi halaman data nasabah untuk aktor Admin KSP. Halaman ini dapat digunakan untuk melihat keseluruhan data pengajuan nasabah berdasarkan preferensinya yang disertai dengan status pinjaman dari tiap pengajuan yang dilakukan nasabah.



Gambar 5. 12 Implementasi Visualisasi Halaman Data Nasabah Pimpinan KSP



Gambar 5. 13 Implementasi Visualisasi Halaman Data Nasabah Admin KSP

BAB 6 PENGUJIAN DAN ANALISIS

Bagian ini memuat tentang pengevaluasian dan pemvalidasian algoritme *data mining* yang telah diimplementasikan dalam bentuk sistem dan juga pengujian sistem yang telah dibangun. Metode yang digunakan untuk mengevaluasi dan memvalidasi algoritme *data mining* yang telah terimplementasi dalam sistem adalah dengan menggunakan *confussion matrix* dan juga kurva ROC. Sedangkan pengujian sistem dilakukan dengan menggunakan *black-box testing* yang bertujuan untuk menguji fungsionalitas sistem. Selain itu juga dilakukan pengujian dari segi kualitas sistem yang telah dibangun dan diujikan langsung kepada responden dari KSP Mitra Raya Wates dengan menggunakan *System Usability Testing* (SUS).

Hasil dari pengevaluasian dan pemvalidasian algoritme *data mining* berikut dengan hasil pengujian *black-box* dan juga uji keberterimaan *user* akan dijabarkan juga pada bab ini dalam bentuk analisis hasil pengujian.

6.1 Evaluasi dan Validasi Algoritme *Data Mining*

Algoritme yang telah diimplementasikan pada sistem perlu dilakukan evaluasi dan validasi algoritme guna memastikan kualitas dan keandalan sistem dalam memprediksi pengajuan kredit. Evaluasi dan validasi algoritme *data mining* dilakukan dengan menggunakan *confussion matrix* dan kurva ROC. Untuk menggunakan *confussion matrix* dan kurva ROC dilakukan dengan menggunakan *data training* dan *data testing* yang telah dibagi pada Bab Perancangan dengan persentase 80% untuk *data training* dan 20% untuk *data testing*. Untuk melakukan evaluasi dan validasi dilakukan dengan memanfaatkan *tool Weka*.

6.1.1 *Confussion Matrix*

Untuk menghasilkan *confussion matrix* maka perlu dilakukan *input data training* dan *data test* pada *Weka*. Secara otomatis proses *classifier C4.5* akan menghasilkan perhitungan *confussion matrix* seperti yang ditampilkan dalam Gambar 6.1. Perhitungan *confussion matrix* dapat dikonversikan ke dalam bentuk tabel seperti yang tertera pada Tabel 6.1.

Tabel 6. 1 Perhitungan *Confussion Matrix*

		Kelas Hasil Prediksi	
		Macet	Lancar
Kelas Asli	Macet	13	1
	Lancar	1	22


```

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances      35          94.5946 %
Incorrectly Classified Instances    2           5.4054 %
Kappa statistic                    0.8851
Mean absolute error                 0.1333
Root mean squared error             0.2282
Relative absolute error             28.5118 %
Root relative squared error         47.0338 %
Total Number of Instances          37

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Cla
          0.929   0.043   0.929     0.929   0.929     0.885   0.969   0.947   MAC
          0.957   0.071   0.957     0.957   0.957     0.885   0.969   0.965   LAN
Weighted Avg.   0.946   0.061   0.946     0.946   0.946     0.885   0.969   0.958

=== Confusion Matrix ===

  a  b  <-- classified as
13  1  |  a = MACET
 1 22  |  b = LANCAR

```

Gambar 6. 1 Hasil Classifier C4.5 Weka

Tabel 6.1 menjelaskan tentang hasil perhitungan *confussion matrix* yang dihasilkan dari proses *classifier C4.5 Weka* dengan menggunakan *data training* 148 records dan *data testing* sebanyak 37 records. Dapat diketahui bahwa kelas yang diklasifikasikan sebagai *true positives* berjumlah 13 records, dimana dapat diartikan bahwa terdapat 13 records kelas macet diklasifikasikan benar sebagai kelas macet. Terdapat satu record diklasifikasikan sebagai *false positives* yang dapat diartikan bahwa satu record dengan kelas asli adalah lancar namun diklasifikasikan salah sebagai kelas macet. Terdapat juga satu record yang diklasifikasikan sebagai *false negatives*, dimana dalam hal ini satu record tersebut bernilai macet namun diklasifikasikan salah sebagai kelas lancar. Dan terdapat 22 records diklasifikasikan sebagai *true negatives*, dimana dapat diartikan bahwa 22 records dengan kelas asli adalah lancar diklasifikasikan benar sebagai kelas lancar. Keseluruhan records berjumlah 37 records yang merupakan *data testing*. Adapun cuplikan *data testing* yang telah mengalami proses klasifikasi ditampilkan pada Tabel 6.2.

Tabel 6. 2 Data Testing Hasil Klasifikasi

JENIS_KELAMIN	TOTAL_PINJAMAN	JUMLAH_TANGGUNGAN	STATUS_PERNIKAHAN	PENDAPATAN_UMUM	TOTAL_BIAYA_HIDUP	'predicted STATUS_PINJAMAN'	STATUS_PINJAMAN
L	RENDAH	SEDANG	MENIKAH	SEDANG	RENDAH	MACET	MACET
P	SANGAT_RENDAH	SEDANG	MENIKAH	SEDANG	RENDAH	LANCAR	LANCAR

JENIS_KELAMIN	TOTAL_PINJAMAN	JUMLAH_TANGGUNGAN	STATUS_PERNIKAHAN	PENDAPATAN_UTAMA	TOTAL_BIAYA_HIDUP	'predicted STATUS_PINJAMAN'	STATUS_PINJAMAN
L	SANGAT_TINGGI	SEDANG	MENIKAH	TINGGI	SEDANG	MACET	MACET
P	RENDAH	SEDIKIT	MENIKAH	SEDANG	RENDAH	LANCAR	LANCAR
L	SANGAT_RENDAH	SEDIKIT	MENIKAH	SEDANG	RENDAH	LANCAR	LANCAR
P	RENDAH	SEDANG	MENIKAH	SEDANG	SEDANG	MACET	MACET
L	SANGAT_RENDAH	SEDANG	MENIKAH	SEDANG	RENDAH	LANCAR	LANCAR
P	RENDAH	SEDIKIT	CERAI	SEDANG	RENDAH	LANCAR	LANCAR
P	RENDAH	SEDANG	MENIKAH	SEDANG	RENDAH	MACET	MACET
P	RENDAH	BANYAK	MENIKAH	SEDANG	RENDAH	MACET	MACET
P	SANGAT_RENDAH	SEDIKIT	CERAI	SEDANG	RENDAH	LANCAR	LANCAR
P	TINGGI	BANYAK	MENIKAH	SEDANG	SEDANG	MACET	MACET
L	TINGGI	SEDANG	MENIKAH	SEDANG	RENDAH	MACET	MACET
L	SANGAT_RENDAH	SEDANG	MENIKAH	SEDANG	RENDAH	LANCAR	LANCAR
L	SEDANG	SEDANG	MENIKAH	SEDANG	RENDAH	MACET	LANCAR
L	SANGAT_RENDAH	SEDIKIT	MENIKAH	SEDANG	RENDAH	LANCAR	LANCAR
L	TINGGI	SEDIKIT	BELUM_MENIKAH	SEDANG	SEDANG	LANCAR	LANCAR
L	RENDAH	SEDANG	MENIKAH	SEDANG	RENDAH	MACET	MACET
L	SANGAT_RENDAH	SEDIKIT	MENIKAH	SEDANG	RENDAH	LANCAR	LANCAR
L	RENDAH	SEDIKIT	BELUM_MENIKAH	SEDANG	RENDAH	LANCAR	LANCAR
P	RENDAH	SEDIKIT	MENIKAH	SEDANG	RENDAH	LANCAR	LANCAR

JENIS_KELAMIN	TOTAL_PINJAMAN	JUMLAH_TANGGUNGAN	STATUS_PERNIKAHAN	PENDAPATAN_UTAMA	TOTAL_BIAYA_HIDUP	'predicted STATUS_PINJAMAN'	STATUS_PINJAMAN
	H		H				
L	TINGGI	SEDIKIT	MENIKAH	SEDANG	RENDAH	LANCAR	MACET
P	RENDAH	SEDIKIT	CERAI	SEDANG	SEDANG	LANCAR	LANCAR
L	RENDAH	SEDIKIT	BELUM_MENIKAH	SEDANG	RENDAH	LANCAR	LANCAR
P	SEDANG	SEDANG	MENIKAH	SEDANG	SEDANG	MACET	MACET
L	SANGAT_RENDAH	SEDIKIT	MENIKAH	SEDANG	RENDAH	LANCAR	LANCAR
L	SANGAT_RENDAH	SEDIKIT	MENIKAH	SEDANG	RENDAH	LANCAR	LANCAR
L	SANGAT_TINGGI	SEDIKIT	BELUM_MENIKAH	SEDANG	RENDAH	LANCAR	LANCAR
L	SANGAT_RENDAH	SEDIKIT	BELUM_MENIKAH	SEDANG	SEDANG	LANCAR	LANCAR
P	RENDAH	SEDIKIT	MENIKAH	SEDANG	RENDAH	LANCAR	LANCAR
L	SANGAT_TINGGI	SEDIKIT	MENIKAH	TINGGI	SEDANG	LANCAR	LANCAR
L	SANGAT_RENDAH	SEDANG	MENIKAH	SEDANG	SEDANG	MACET	MACET
L	SEDANG	SEDANG	MENIKAH	SEDANG	RENDAH	MACET	MACET
L	SANGAT_TINGGI	SEDANG	MENIKAH	SEDANG	RENDAH	MACET	MACET
L	SANGAT_RENDAH	SEDIKIT	MENIKAH	SEDANG	RENDAH	LANCAR	LANCAR
L	RENDAH	SEDIKIT	MENIKAH	SEDANG	RENDAH	LANCAR	LANCAR
P	SEDANG	SEDANG	MENIKAH	SEDANG	RENDAH	MACET	MACET

Pada Tabel 6.2, dapat diketahui data yang masuk *false positives* dan *false negatives* yang telah diberi *highlight* dengan warna *cell* kuning. Data yang masuk

pada *false positives* ditampilkan pada Tabel 6.3 mengalami kesalahan klasifikasi dikarenakan berdasarkan *rule* yang telah dibuat pada proses *learning*, atribut JUMLAH_TANGGUNGAN adalah SEDANG dan TOTAL_PINJAMAN adalah SEDANG maka seharusnya STATUS_PINJAMAN akan MACET. Sedangkan data yang masuk pada *false negatives*, yang ditampilkan pada Tabel 6.4 mengalami kesalahan klasifikasi dikarenakan berdasarkan *rule* yang telah dibuat, atribut JUMLAH_TANGGUNGAN adalah SEDIKIT dan PENDAPATAN_UTAMA adalah SEDANG dan TOTAL_BIAYA_HIDUP adalah RENDAH, maka STATUS_PINJAMAN akan LANCAR.

Tabel 6. 3 Data Testing Hasil Klasifikasi pada False Positives

JENIS_KELAMIN	TOTAL_PINJAMAN	JUMLAH_TANGGUNGAN	STATUS_PERNIKAHAN	PENDAPATAN_UTAMA	TOTAL_BIAYA_HIDUP	'predicted STATUS_PINJAMAN'	STATUS_PINJAMAN
L	SEDANG	SEDANG	MENIKAH	SEDANG	RENDAH	MACET	LANCAR

Tabel 6. 4 Data Testing Hasil Klasifikasi pada False Negatives

JENIS_KELAMIN	TOTAL_PINJAMAN	JUMLAH_TANGGUNGAN	STATUS_PERNIKAHAN	PENDAPATAN_UTAMA	TOTAL_BIAYA_HIDUP	'predicted STATUS_PINJAMAN'	STATUS_PINJAMAN
L	TINGGI	SEDIKIT	MENIKAH	SEDANG	RENDAH	LANCAR	MACET

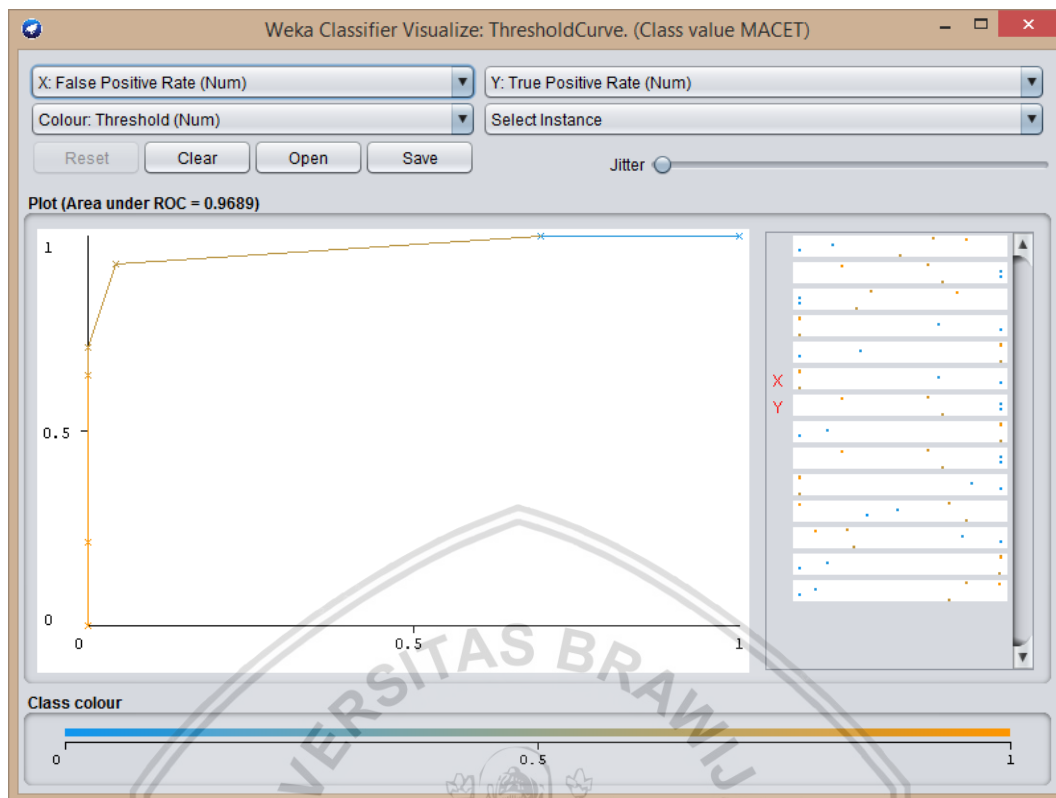
Validitas sistem dapat dinilai dengan cara menghitung akurasi berdasarkan tabel matriks *confussion matrix*. Untuk menghitung tingkat akurasi digunakan Persamaan 2.3.

$$Akurasi = \frac{13 + 22}{13 + 22 + 1 + 1} = \frac{35}{37} = 0.945946$$

Dihasilkan nilai 0.945946 dimana nilai tersebut sama dengan nilai yang dihasilkan dari proses *classifier* C4.5 *Weka* yaitu 94,5946%. Dapat diartikan bahwa tingkat akurasi sistem sangat baik karena mendekati 100%.

6.1.2 Kurva ROC

Sedangkan kurva ROC yang dihasilkan dari proses *classifiers* menggunakan *Weka* dengan pembagian *data training* berjumlah 148 *records* dan *data testing* berjumlah 37 *records* dapat dilihat dalam Gambar 6.2.



Gambar 6. 2 Hasil *Threshold Curve Class Value Macet*

Dalam Gambar 6.2 dapat diketahui bahwa nilai kurva ROC yang dihasilkan untuk *class value* macet bernilai 0,9689. Sehingga dengan merujuk pada pengklasifikasian nilai AUC berdasarkan Gorunescu (2011), klasifikasi C4.5 untuk prediksi kredit macet berdasarkan preferensi nasabah pada KSP Mitra Raya Wates dapat digolongkan sebagai klasifikasi yang sangat baik karena berada pada nilai AUC 0.90 – 1.00.

6.2 Pengujian *Black Box –Validation Testing*

Pada tahap ini dilakukan pengujian persyaratan fungsionalitas sistem yang telah dibangun menggunakan *black-box validation testing*. Pengujian dilakukan dengan cara mendefinisikan kasus uji berdasarkan *scenario use case*, dan dilakukan perbandingan antara hasil dengan ekspektasi pengujian.

6.2.1 Pengujian Fitur Memasukkan *Dataset*

Pengujian fitur memasukkan *dataset* dilakukan berdasarkan kasus uji pada *basic flow* dan *alt1*.

Tabel 6. 5 Kasus Uji Memasukkan *Dataset*: *Basic Flow*

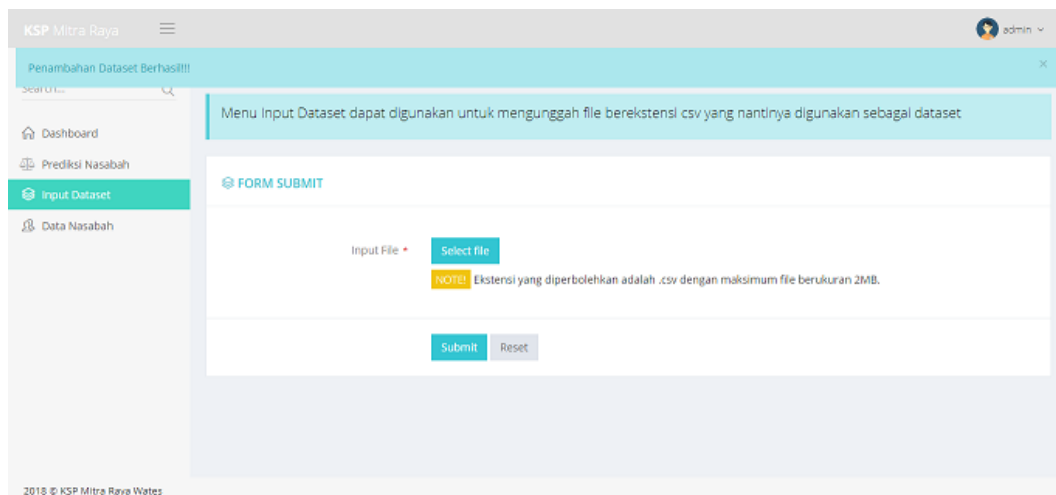
Test Case Code	BB-001
Test Case Name	Test Case memasukkan <i>dataset</i>
Code Use Case	UC-SRPK-1

Testing Objectives	Pengujian ini dilakukan untuk memastikan Pimpinan KSP atau Admin KSP dapat mengunggah <i>dataset</i> ke sistem
Test Case	Mengunggah <i>dataset</i> dengan ekstensi csv dengan ukuran di bawah 2MB
Testing Step	<ol style="list-style-type: none"> 1. Penguji mengakses menu <i>input dataset</i> 2. Penguji memilih <i>dataset</i> yang ingin diunggah 3. Penguji menekan tombol <i>submit</i>
Expectation	Menampilkan pesan <i>dataset</i> berhasil di- <i>upload</i>
Result	Sistem menampilkan pesan <i>dataset</i> berhasil di- <i>upload</i>
Status	Valid

Tabel 6.5 menjelaskan kasus uji *black-box* untuk fitur memasukkan *dataset*. Kasus uji dilakukan dengan mengunggah *dataset* dengan ekstensi csv dengan ukuran di bawah 2MB. Hasil pengujian yang telah dilakukan menunjukkan status valid. Gambar 6.4 dan Gambar 6.5 menunjukkan hasil dari pengujian pada kasus dengan *test case code* BB-001.



Gambar 6. 3 Hasil Pengujian Fitur Memasukkan *Dataset* Pimpinan KSP: *Basic Flow*



Gambar 6. 4 Hasil Pengujian Fitur Memasukkan *Dataset* Admin KSP: *Basic Flow*

Tabel 6. 6 Kasus Uji Memasukkan *Dataset*: alt1

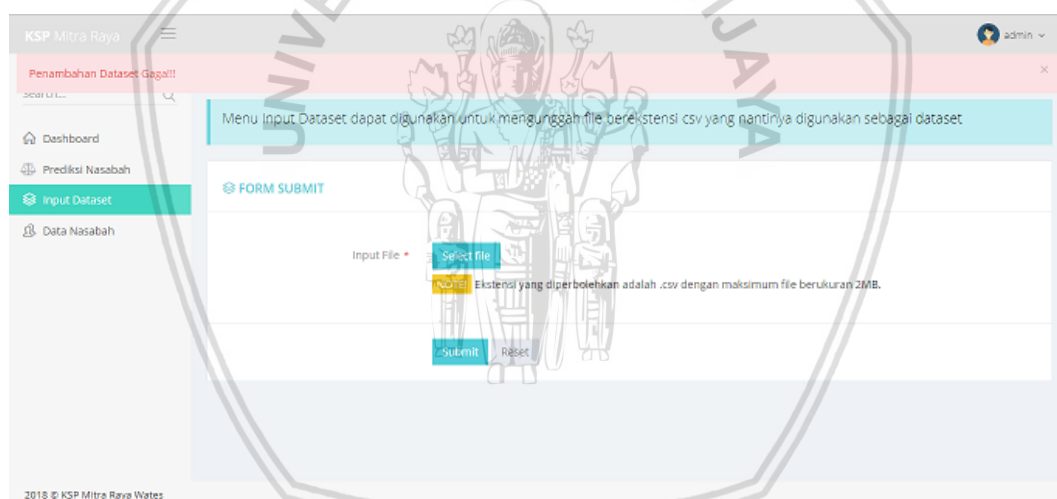
Test Case Code	BB-002
Test Case Name	Test Case memasukkan <i>dataset</i>
Code Use Case	UC-SRPK-1
Testing Objectives	Pengujian ini dilakukan untuk memastikan Pimpinan KSP atau Admin KSP dapat mengunggah <i>dataset</i> ke sistem
Test Case	Mengunggah <i>dataset</i> dengan ekstensi selain <i>csv</i> atau dengan ukuran di atas 2MB
Testing Step	<ol style="list-style-type: none"> 1. Penguji mengakses menu <i>input dataset</i> 2. Penguji memilih <i>dataset</i> yang ingin diunggah 3. Penguji menekan tombol <i>submit</i>
Expectation	Menampilkan pesan <i>dataset</i> berhasil di- <i>upload</i>
Result	Sistem menampilkan pesan <i>dataset</i> gagal di- <i>upload</i>
Status	Valid

Tabel 6.6 menjelaskan kasus uji *black-box* untuk fitur memasukkan *dataset*. Kasus uji dilakukan dengan mengunggah *dataset* dengan ekstensi selain *csv* atau

dengan ukuran di atas 2MB. Hasil pengujian yang telah dilakukan menunjukan status valid. Gambar 6.6 dan Gambar 6.7 menunjukkan hasil dari pengujian pada kasus dengan *test case code* BB-002.



Gambar 6. 5 Hasil Pengujian Fitur Memasukkan *Dataset* Pimpinan KSP: *alt1*



Gambar 6. 6 Hasil Pengujian Fitur Memasukkan *Dataset* Admin KSP: *alt1*

6.2.2 Pengujian Fitur Melihat Data Pengajuan Nasabah

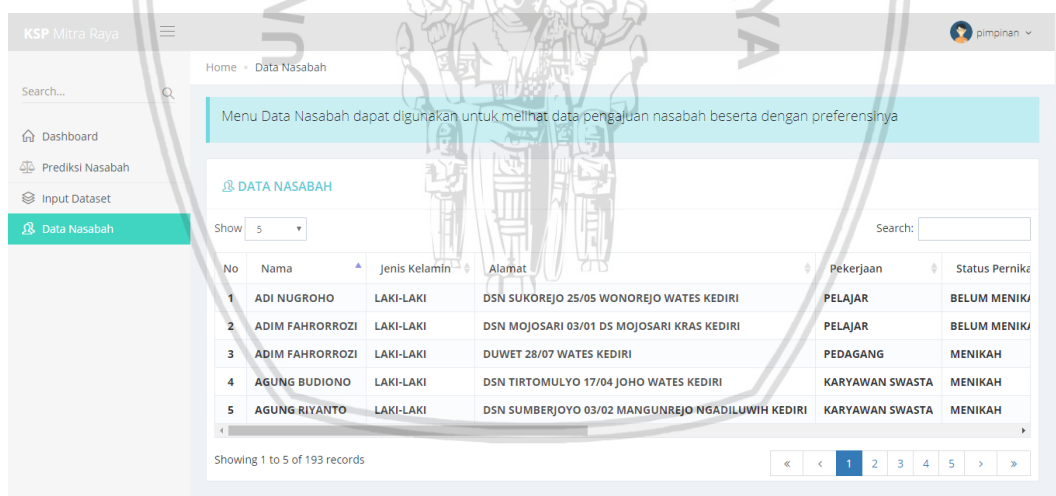
Pengujian fitur melihat data nasabah dilakukan berdasarkan kasus uji pada *basic flow*.

Tabel 6. 7 Kasus Uji Melihat Data Nasabah: *Basic Flow*

Test Case Code	BB-003
Test Case Name	Test Case melihat data pengajuan nasabah
Code Use Case	UC-SRPK-2

Testing Objectives	Pengujian ini dilakukan untuk memastikan Pimpinan KSP atau Admin KSP dapat melihat data pengajuan nasabah dari KSP Mitra Raya
Test Case	Menekan tombol menu data nasabah
Testing Step	1. Penguji mengakses menu data nasabah
Expectation	Menampilkan data pengajuan nasabah dalam bentuk <i>table</i>
Result	Sistem menampilkan data pengajuan nasabah dalam bentuk table
Status	Valid

Tabel 6.7 menjelaskan kasus uji *black-box* untuk fitur melihat data pengajuan nasabah. Kasus uji dilakukan dengan mengakses menu data nasabah. Hasil pengujian yang telah dilakukan menunjukkan status valid. Gambar 6.8 dan Gambar 6.9 menunjukkan hasil dari pengujian pada kasus dengan *test case code* BB-003.



Gambar 6. 7 Hasil Pengujian Fitur Melihat Nasabah Pimpinan KSP: *Basic Flow*

KSP Mitra Raya

Home - Data Nasabah

Search...

Dashboard

Lihat Akurasi

Lihat Tree

Input Dataset

Data Nasabah

Menu Data Nasabah dapat digunakan untuk melihat data pengajuan nasabah beserta dengan preferensinya

DATA NASABAH

Show: 5 Search:

No	Nama	Jenis Kelamin	Alamat	Pekerjaan	Status Pernikahan
1	ADI NUGROHO	LAKI-LAKI	DSN SUKOREJO 25/05 WONOREJO WATES KEDIRI	PELAJAR	BELUM MENIKAH
2	ADIM FAHRORROZI	LAKI-LAKI	DSN MOJOSARI 03/01 DS MOJOSARI KRAS KEDIRI	PELAJAR	BELUM MENIKAH
3	ADIM FAHRORROZI	LAKI-LAKI	DUWET 28/07 WATES KEDIRI	PEDAGANG	MENIKAH
4	AGUNG BUDIONO	LAKI-LAKI	DSN TIRTOMULYO 17/04 JOHO WATES KEDIRI	KARYAWAN SWASTA	MENIKAH
5	AGUNG RIYANTO	LAKI-LAKI	DSN SUMBERJOYO 03/02 MANGUNREJO NGADILUWIH KEDIRI	KARYAWAN SWASTA	MENIKAH

Showing 1 to 5 of 193 records

Gambar 6. 8 Hasil Pengujian Fitur Melihat Nasabah Admin KSP: *Basic Flow*

6.2.3 Pengujian Fitur Memasukkan Data Pengaju Pinjaman

Pengujian fitur memasukkan data pengaju pinjaman dilakukan berdasarkan kasus uji pada *basic flow* dan *alt1*.

Tabel 6. 8 Kasus Uji Memasukkan Data Pengaju Pinjaman: *Basic Flow*

Test Case Code	BB-004
Test Case Name	Test Case memasukkan data pengaju pinjaman
Code Use Case	UC-SRPK-3
Testing Objectives	Pengujian ini dilakukan untuk memastikan Pimpinan KSP dapat memasukan data pengaju pinjaman
Test Case	Mengisi <i>form</i> pengajuan pinjaman nasabah dengan setiap <i>field</i> terisi.
Testing Step	<ol style="list-style-type: none"> 1. Penguji mengakses menu prediksi nasabah 2. Penguji mengisi setiap <i>field</i> berdasarkan dari informasi nasabah pengaju pinjaman 3. Penguji menekan tombol submit
Expectation	Menampilkan hasil prediksi nasabah
Result	Sistem menampilkan hasil prediksi nasabah
Status	Valid

Tabel 6.8 menjelaskan kasus uji *black-box* untuk fitur memasukkan data pengaju pinjaman. Kasus uji dilakukan dengan mengisi *form* pengajuan pinjaman nasabah dengan setiap *field* terisi. Hasil pengujian yang telah dilakukan menunjukkan status valid. Gambar 6.10 menunjukkan hasil dari pengujian pada kasus dengan *test case code* BB-004.

KSP Mitra Raya

Home • Prediksi Nasabah • Submit Data • Hasil Prediksi

Search...

Dashboard

Prediksi Nasabah

Input Dataset

Data Nasabah

HASIL PREDIKSI

Nama:	SADI
Jenis Kelamin:	L
Total Pinjaman:	5000000
Jumlah Tanggungan:	2
Status Pernikahan:	BELUM_MENIKAH
Pendapatan Utama:	2500000
Total Biaya Hidup:	1500000
Hasil Prediksi:	LANCAR

2018 © KSP Mitra Raya Wates

Gambar 6. 9 Hasil Pengujian Fitur Memasukkan Data Pengaju Pinjaman: *Basic Flow*

Tabel 6. 9 Kasus Uji Memasukkan Data Pengaju Pinjaman: alt1

Test Case Code	BB-005
Test Case Name	Test Case memasukkan data pengaju pinjaman
Code Use Case	UC-SRPK-3
Testing Objectives	Pengujian ini dilakukan untuk memastikan Pimpinan KSP dapat memasukan data pengaju pinjaman
Test Case	Mengisi <i>form</i> pengajuan pinjaman nasabah dengan ada salah satu <i>field</i> tidak terisi.
Testing Step	<ol style="list-style-type: none"> 1. Penguji mengakses menu prediksi nasabah 2. Penguji mengisikan setiap <i>field</i> berdasarkan dari informasi nasabah pengaju pinjaman 3. Penguji menekan tombol submit
Expectation	Menampilkan pesan isi bagian formulir yang kosong

Result	Sistem menampilkan pesan isi bagian formulir yang kosong
Status	Valid

Tabel 6.9 menjelaskan kasus uji *black-box* untuk fitur memasukkan data pengaju pinjaman. Kasus uji dilakukan dengan mengisi *form* pengajuan pinjaman nasabah dengan salah satu *field* tidak terisi. Hasil pengujian yang telah dilakukan menunjukkan status valid. Gambar 6.11 menunjukkan hasil dari pengujian pada kasus dengan *test case code* BB-005.

Gambar 6. 10 Hasil Pengujian Fitur Memasukkan Data Pengaju Pinjaman: *alt1*

6.2.4 Pengujian Fitur Melihat Jumlah Pengajuan

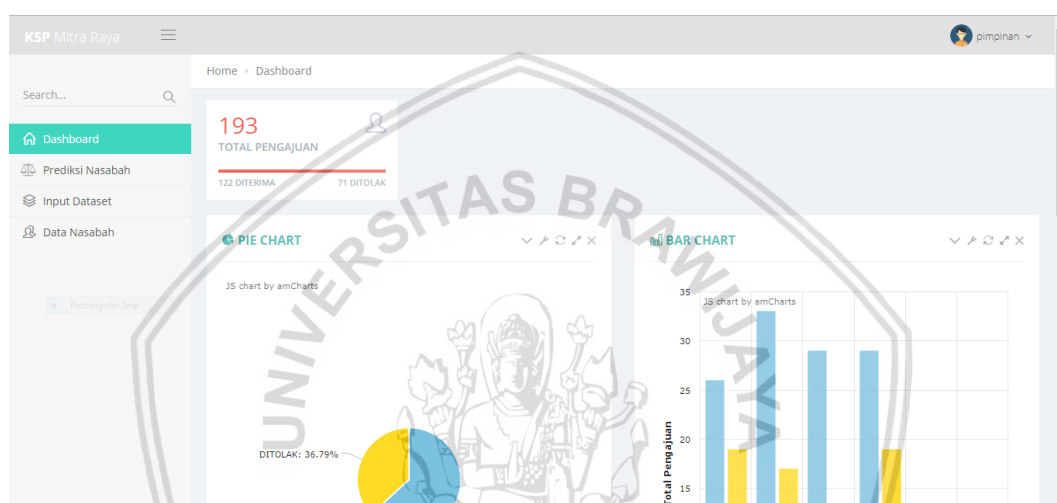
Pengujian fitur melihat jumlah pengajuan dilakukan berdasarkan kasus uji pada *basic flow*.

Tabel 6. 10 Kasus Uji Melihat Jumlah Pengajuan: *Basic Flow*

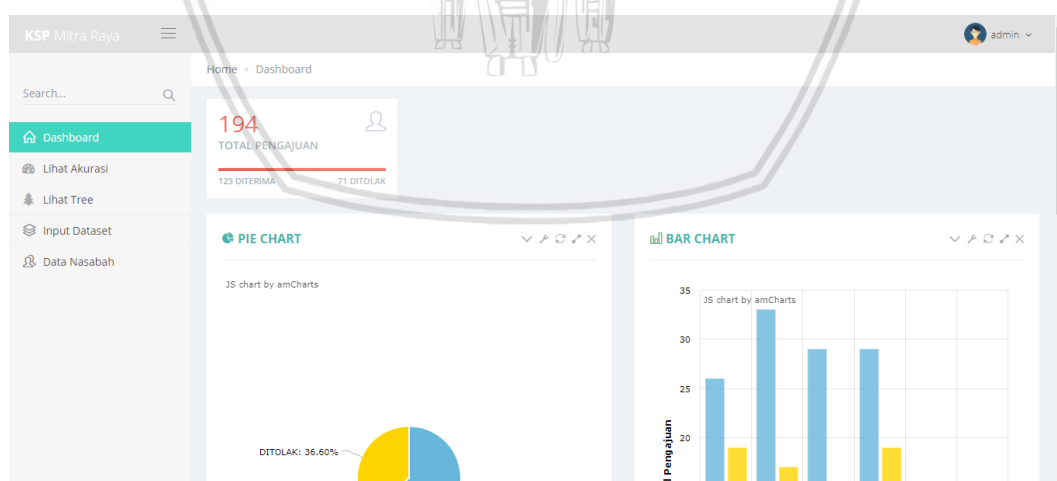
Test Case Code	BB-006
Test Case Name	<i>Test Case</i> melihat jumlah pengajuan
Code Use Case	UC-SRPK-5
Testing Objectives	Pengujian ini dilakukan untuk memastikan Pimpinan KSP atau Admin KSP dapat melihat jumlah pengajuan
Test Case	Mengakses <i>menu dashboard</i>
Testing Step	1. Penguji mengakses <i>menu dashboard</i>
Expectation	Menampilkan jumlah pengajuan, baik pengajuan ditolak maupun diterima dalam bentuk angka

Result	Sistem menampilkan jumlah pengajuan, baik pengajuan ditolak maupun diterima dalam bentuk angka
Status	Valid

Tabel 6.10 menjelaskan kasus uji *black-box* untuk fitur melihat jumlah pengajuan. Kasus uji dilakukan dengan mengakses menu *dashboard*. Hasil pengujian yang telah dilakukan menunjukkan status valid. Gambar 6.12 dan Gambar 6.13 menunjukkan hasil dari pengujian pada kasus dengan *test case code* BB-006.



Gambar 6. 11 Hasil Pengujian Fitur Melihat Jumlah Pengajuan Pimpinan KSP: Basic Flow



Gambar 6. 12 Hasil Pengujian Fitur Melihat Jumlah Pengajuan Admin KSP: Basic Flow

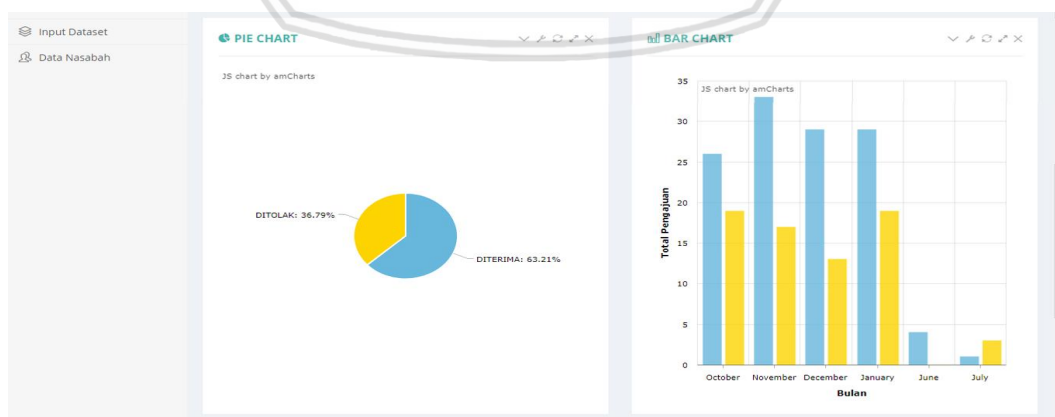
6.2.5 Pengujian Fitur Melihat Persentase Pengajuan

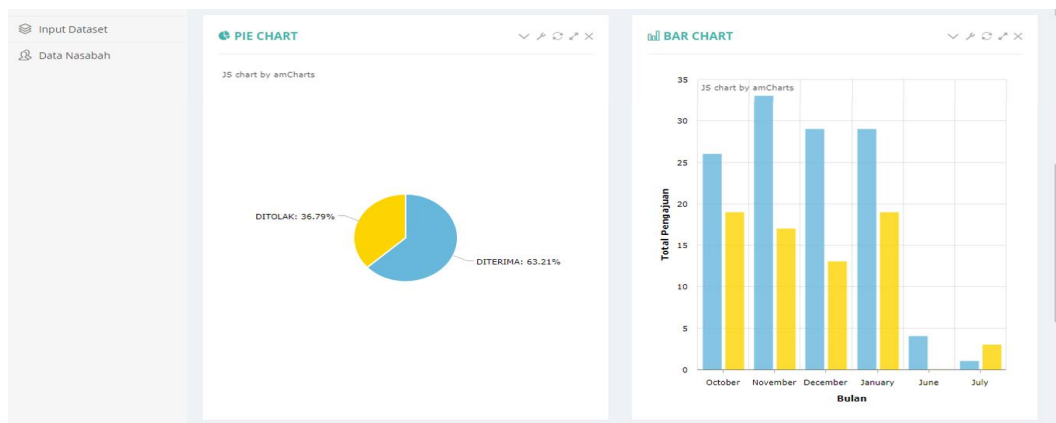
Pengujian fitur melihat persentase pengajuan dilakukan berdasarkan kasus uji pada *basic flow*.

Tabel 6. 11 Kasus Uji Melihat Persentase Pengajuan: *Basic Flow*

Test Case Code	BB-007
Test Case Name	Test Case melihat persentase pengajuan
Code Use Case	UC-SRPK-6
Testing Objectives	Pengujian ini dilakukan untuk memastikan Pimpinan KSP atau Admin KSP dapat melihat persentase pengajuan dalam bentuk <i>pie chart</i>
Test Case	Mengakses <i>menu dashboard</i>
Testing Step	1. Penguji mengakses <i>menu dashboard</i>
Expectation	Menampilkan persentase pengajuan, baik pengajuan ditolak maupun diterima dalam bentuk <i>pie chart</i>
Result	Sistem menampilkan persentase pengajuan, baik pengajuan ditolak maupun diterima dalam bentuk <i>pie chart</i>
Status	Valid

Tabel 6.11 menjelaskan kasus uji *black-box* untuk fitur melihat jumlah pengajuan. Kasus uji dilakukan dengan mengakses menu *dashboard*. Hasil pengujian yang telah dilakukan menunjukkan status valid. Gambar 6.14 dan Gambar 6.15 menunjukkan hasil dari pengujian pada kasus dengan *test case code* BB-007.

Gambar 6. 13 Hasil Pengujian Fitur Melihat Persentase Pengajuan Pimpinan KSP: *Basic Flow*



Gambar 6. 14 Hasil Pengujian Fitur Melihat Persentase Pengajuan Admin KSP: Basic Flow

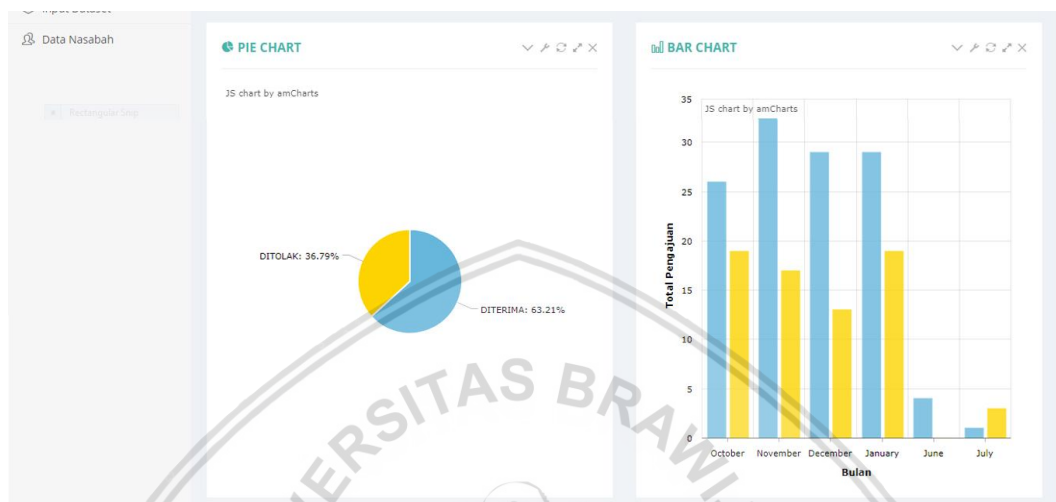
6.2.6 Pengujian Fitur Melihat *Time-Series* Pengajuan

Pengujian fitur melihat *time-series* pengajuan dilakukan berdasarkan kasus uji pada *basic flow*.

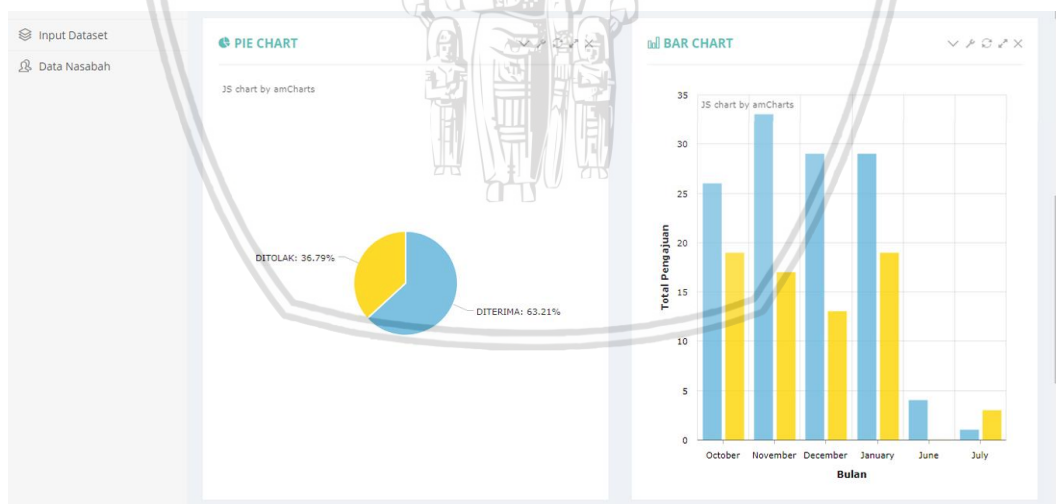
Tabel 6. 12 Kasus Uji Melihat *Time-Series* Pengajuan: Basic Flow

Test Case Code	BB-008
Test Case Name	Test Case melihat <i>time-series</i> pengajuan
Code Use Case	UC-SRPK-7
Testing Objectives	Pengujian ini dilakukan untuk memastikan Pimpinan KSP atau Admin KSP dapat melihat <i>time-series</i> pengajuan dalam bentuk <i>bar chart</i>
Test Case	Mengakses <i>menu dashboard</i>
Testing Step	1. Penguji mengakses <i>menu dashboard</i>
Expectation	Menampilkan <i>time-series</i> pengajuan, baik pengajuan ditolak maupun diterima dalam bentuk <i>bar chart</i>
Result	Sistem menampilkan <i>time-series</i> pengajuan, baik pengajuan ditolak maupun diterima dalam bentuk <i>bar chart</i>
Status	Valid

Tabel 6.12 menjelaskan kasus uji *black-box* untuk fitur melihat *time-series* pengajuan. Kasus uji dilakukan dengan mengakses menu *dashboard*. Hasil pengujian yang telah dilakukan menunjukkan status valid. Gambar 6.16 dan Gambar 6.17 menunjukkan hasil dari pengujian pada kasus dengan *test case code* BB-008.



Gambar 6. 15 Hasil Pengujian Fitur Melihat *Time-Series* Pengajuan Pimpinan KSP: *Basic Flow*



Gambar 6. 16 Hasil Pengujian Fitur Melihat *Time-Series* Pengajuan Admin KSP: *Basic Flow*

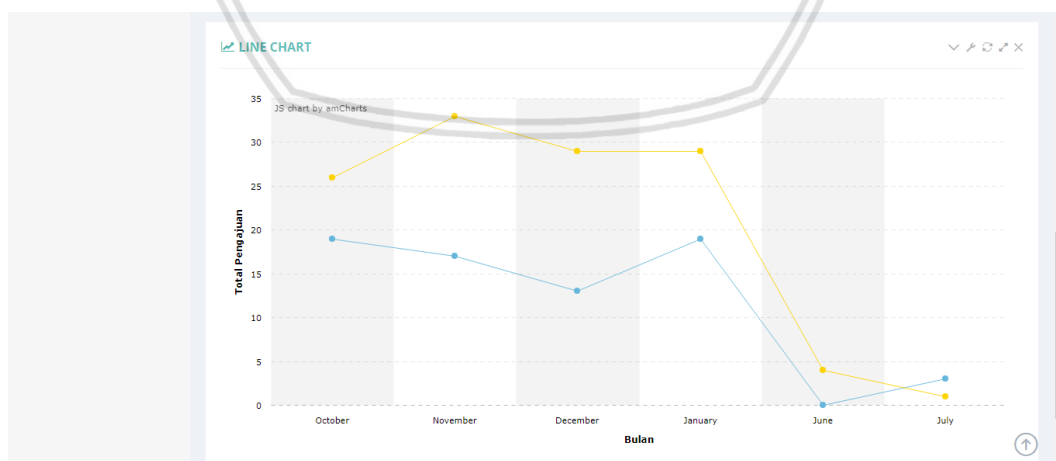
6.2.7 Pengujian Fitur Melihat *Trend* Pengajuan

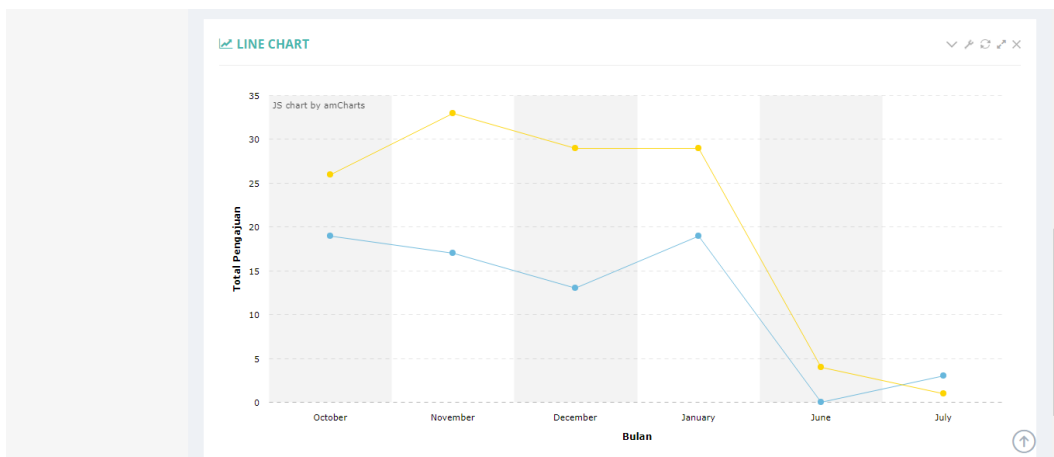
Pengujian fitur melihat *trend* pengajuan dilakukan berdasarkan kasus uji pada *basic flow*.

Tabel 6. 13 Kasus Uji Melihat *Trend* Pengajuan: *Basic Flow*

Test Case Code	BB-009
Test Case Name	Test Case melihat <i>trend</i> pengajuan
Code Use Case	UC-SRPK-8
Testing Objectives	Pengujian ini dilakukan untuk memastikan Pimpinan KSP atau Admin KSP dapat melihat <i>trend</i> pengajuan dalam bentuk <i>line chart</i>
Test Case	Mengakses <i>menu dashboard</i>
Testing Step	1. Penguji mengakses <i>menu dashboard</i>
Expectation	Menampilkan <i>trend</i> pengajuan, baik pengajuan ditolak maupun diterima dalam bentuk <i>line chart</i>
Result	Sistem menampilkan <i>trend</i> pengajuan, baik pengajuan ditolak maupun diterima dalam bentuk <i>line chart</i>
Status	Valid

Tabel 6.13 menjelaskan kasus uji *black-box* untuk fitur melihat *trend* pengajuan. Kasus uji dilakukan dengan mengakses menu *dashboard*. Hasil pengujian yang telah dilakukan menunjukkan status valid. Gambar 6.18 dan Gambar 6.19 menunjukkan hasil dari pengujian pada kasus dengan *test case code* BB-009.

Gambar 6. 17 Hasil Pengujian Fitur Melihat *Trend* Pengajuan Pimpinan KSP: *Basic Flow*



Gambar 6. 18 Hasil Pengujian Fitur Melihat *Trend* Pengajuan Admin KSP: *Basic Flow*

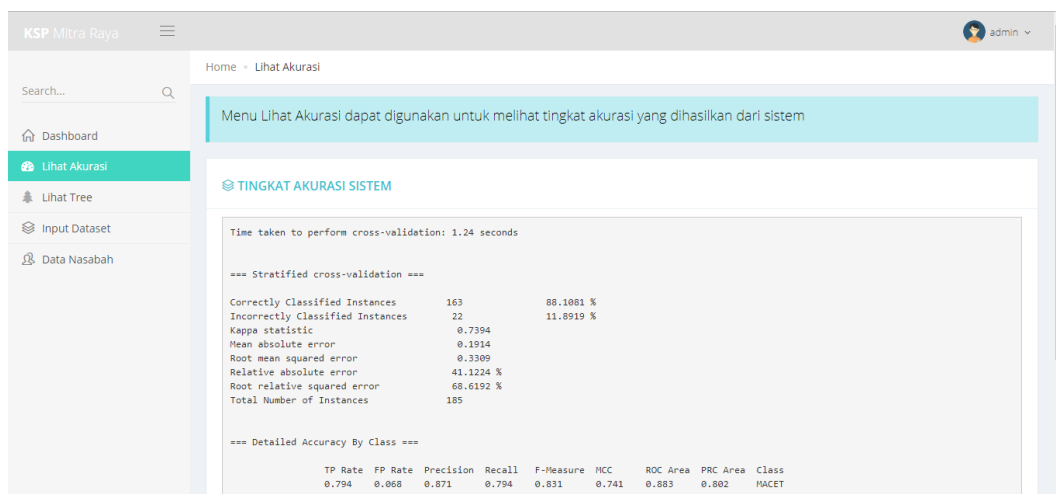
6.2.8 Pengujian Fitur Melihat Tingkat Akurasi

Pengujian fitur melihat tingkat akurasi dilakukan berdasarkan kasus uji pada *basic flow*.

Tabel 6. 14 Kasus Uji Melihat Tingkat Akurasi: *Basic Flow*

Test Case Code	BB-010
Test Case Name	Test Case melihat tingkat akurasi
Code Use Case	UC-SRPK-9
Testing Objectives	Pengujian ini dilakukan untuk memastikan Admin KSP dapat melihat tingkat akurasi
Test Case	Mengakses menu lihat akurasi
Testing Step	1. Penguji mengakses menu lihat akurasi
Expectation	Menampilkan tingkat akurasi sistem
Result	Sistem menampilkan tingkat akurasi sistem
Status	Valid

Tabel 6.14 menjelaskan kasus uji *black-box* untuk fitur melihat tingkat akurasi. Kasus uji dilakukan dengan mengakses menu lihat akurasi. Hasil pengujian yang telah dilakukan menunjukkan status valid. Gambar 6.20 menunjukkan hasil dari pengujian pada kasus dengan *test case code* BB-010.



Gambar 6. 19 Hasil Pengujian Fitur Melihat Tingkat Akurasi: *Basic Flow*

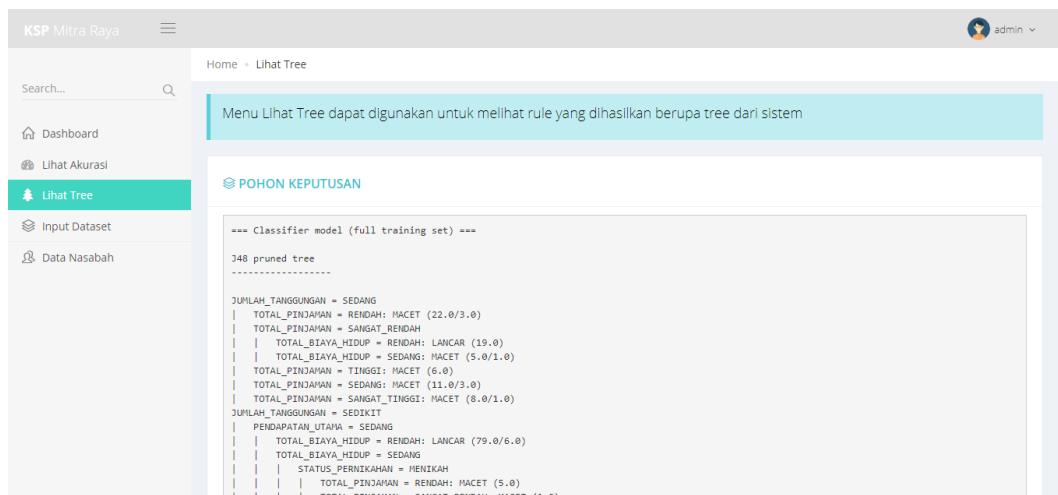
6.2.9 Pengujian Fitur Melihat Pohon Keputusan

Pengujian fitur melihat pohon keputusan dilakukan berdasarkan kasus uji pada *basic flow*.

Tabel 6. 15 Kasus Uji Melihat Pohon Keputusan: *Basic Flow*

Test Case Code	BB-011
Test Case Name	Test Case melihat pohon keputusan
Code Use Case	UC-SRPK-10
Testing Objectives	Pengujian ini dilakukan untuk memastikan Admin KSP dapat melihat pohon keputusan
Test Case	Mengakses menu lihat <i>tree</i>
Testing Step	1. Penguji mengakses menu lihat <i>tree</i>
Expectation	Menampilkan pohon keputusan
Result	Sistem menampilkan pohon keputusan
Status	Valid

Tabel 6.15 menjelaskan kasus uji *black-box* untuk fitur melihat pohon keputusan. Kasus uji dilakukan dengan mengakses menu lihat *tree*. Hasil pengujian yang telah dilakukan menunjukkan status valid. Gambar 6.21 menunjukkan hasil dari pengujian pada kasus dengan *test case code* BB-011.



Gambar 6. 20 Hasil Pengujian Fitur Melihat Pohon Keputusan: *Basic Flow*

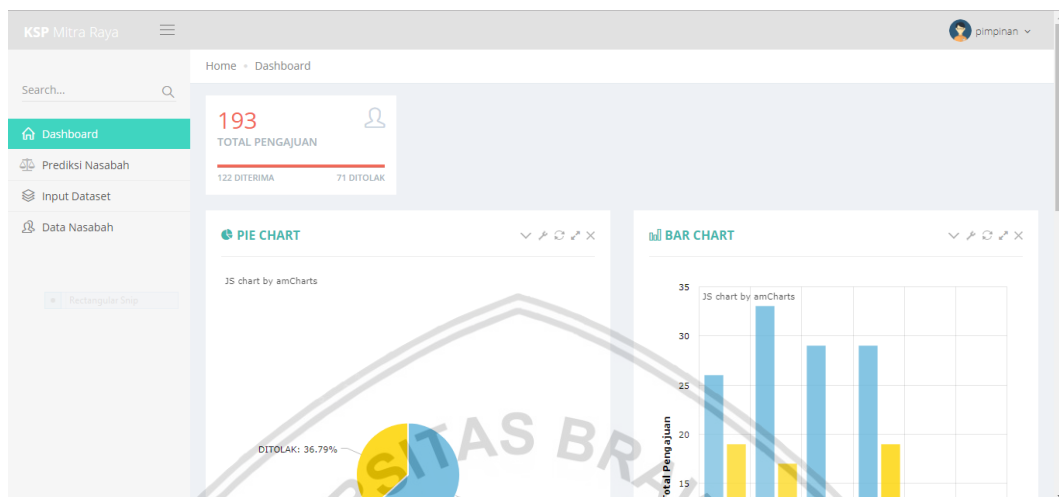
6.2.10 Pengujian Fitur Autentifikasi

Pengujian fitur melihat autentifikasi dilakukan berdasarkan kasus uji pada *basic flow* dan *alt1*.

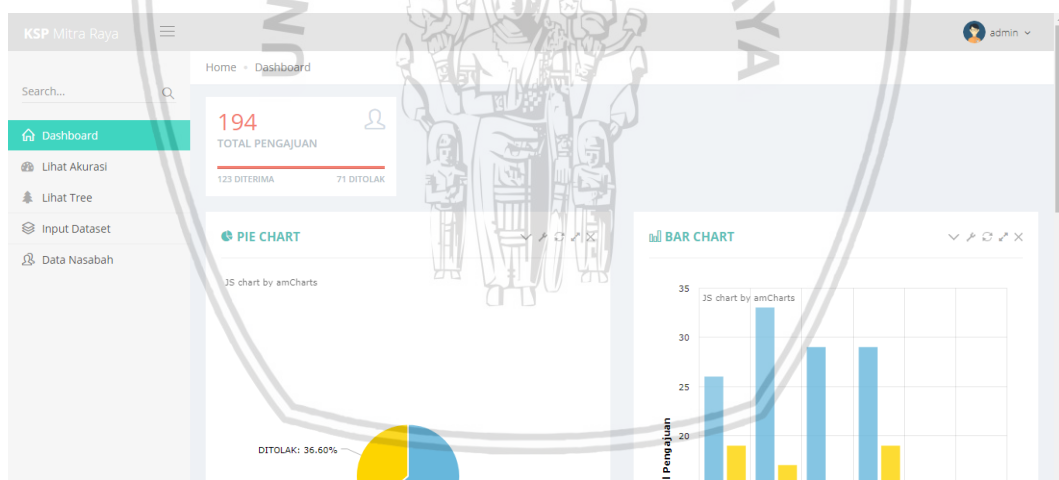
Tabel 6. 16 Kasus Uji Autentifikasi: *Basic Flow*

Test Case Code	BB-012
Test Case Name	Test Case autentifikasi
Code Use Case	UC-SRPK-11
Testing Objectives	Pengujian ini dilakukan untuk memastikan Pimpinan KSP atau Admin KSP dapat masuk sebagai pengguna ke sistem
Test Case	Mengisikan <i>username</i> dan <i>password</i> sesuai dengan data yang telah didaftarkan di <i>database</i>
Testing Step	<ol style="list-style-type: none"> 1. Penguji mengakses <i>menu login</i> 2. Penguji memasukkan <i>username</i> dan <i>password</i> 3. Penguji menekan tombol login
Expectation	Menampilkan halaman <i>dashboard</i>
Result	Sistem menampilkan halaman <i>dashboard</i>
Status	Valid

Tabel 6.16 menjelaskan kasus uji *black-box* untuk fitur autentifikasi. Kasus uji dilakukan dengan mengisikan *username* dan *password* sesuai dengan data yang telah didaftarkan di *database*. Hasil pengujian yang telah dilakukan menunjukkan status valid. Gambar 6.22 dan Gambar 6.23 menunjukkan hasil dari pengujian pada kasus dengan *test case code* BB-012.



Gambar 6. 21 Hasil Pengujian Fitur Autentifikasi Pimpinan KSP: *Basic Flow*



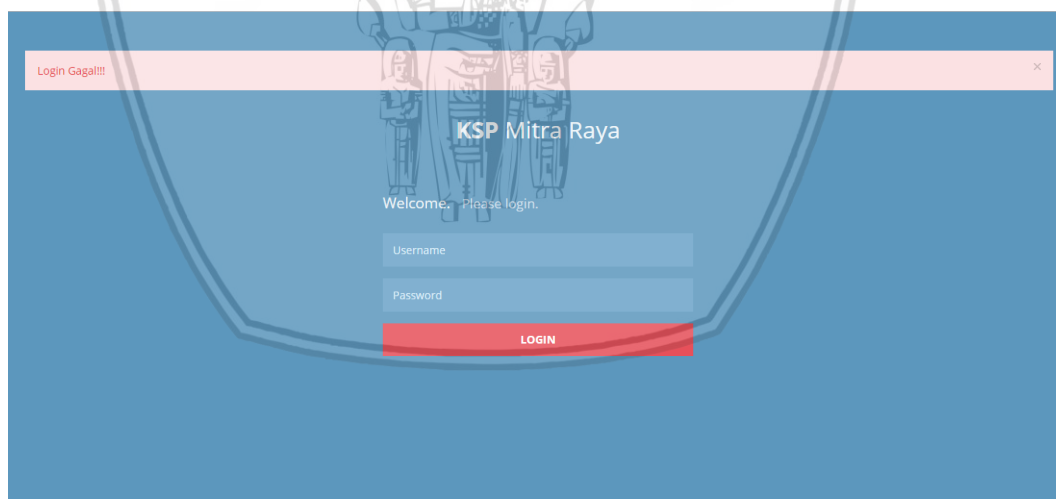
Gambar 6. 22 Hasil Pengujian Fitur Autentifikasi Admin KSP: *Basic Flow*

Tabel 6. 17 Kasus Uji Autentifikasi: *alt1*

Test Case Code	BB-013
Test Case Name	Test Case autentifikasi
Code Use Case	UC-SRPK-11
Testing Objectives	Pengujian ini dilakukan untuk memastikan Pimpinan KSP atau Admin KSP dapat masuk sebagai pengguna ke

	sistem
Test Case	Mengisikan <i>username</i> dan <i>password</i> dengan data yang belum didaftarkan di <i>database</i>
Testing Step	<ol style="list-style-type: none"> 1. Penguji mengakses <i>menu login</i> 2. Penguji memasukkan <i>username</i> dan <i>password</i> 3. Penguji menekan tombol login
Expectation	Menampilkan pesan <i>error</i> jika <i>username</i> dan <i>password</i> salah
Result	Sistem menampilkan pesan <i>error</i> jika <i>username</i> dan <i>password</i> salah
Status	Valid

Tabel 6.17 menjelaskan kasus uji *black-box* untuk fitur autentifikasi. Kasus uji dilakukan dengan mengisikan *username* dan *password* dengan data yang belum didaftarkan di *database*. Hasil pengujian yang telah dilakukan menunjukkan status valid. Gambar 6.16 menunjukkan hasil dari pengujian pada kasus dengan *test case code* BB-013.



Gambar 6. 23 Hasil Pengujian Fitur Autentifikasi: *alt1*

6.2.11 Pengujian Fitur Keluar

Pengujian keluar dilakukan berdasarkan kasus uji pada *basic flow*.

Tabel 6. 18 Kasus Uji Keluar: *Basic Flow*

Test Case Code	BB-014
Test Case Name	Test Case keluar
Code Use Case	UC-SRPK-12
Testing Objectives	Pengujian ini dilakukan untuk memastikan Pimpinan KSP atau Admin KSP dapat keluar dari sistem
Test Case	Mengakses <i>menu logout</i>
Testing Step	1. Penguji mengakses <i>menu logout</i>
Expectation	Menampilkan halaman <i>login</i>
Result	Sistem menampilkan halaman <i>login</i>
Status	Valid

Tabel 6.18 menjelaskan kasus uji *black-box* untuk fitur keluar. Kasus uji dilakukan dengan mengakses *menu logout*. Hasil pengujian yang telah dilakukan menunjukkan status valid. Gambar 6.25 menunjukkan hasil dari pengujian pada kasus dengan *test case code* BB-014.



Gambar 6. 24 Hasil Pengujian Fitur Autentifikasi: *Basic Flow*

Dari beberapa pengujian yang telah dilakukan berdasarkan *use case scenario*, didapatkan hasil bahwa semua *use case* telah tervalidasi. Sehingga dapat

diartikan bahwa sistem yang dibangun siap digunakan oleh KSP Mitra Raya Wates dan siap untuk dilakukan uji keberterimaan *user*.

6.3 Pengujian *Black Box – Usability Testing*

Tahap ini dilakukan proses *usability testing* guna mengetahui kualitas sistem yang dibuat. Metode yang digunakan dalam *usability testing* adalah dengan menggunakan SUS yang terdiri dari 10 item pernyataan dengan nilai skala antara satu sampai lima. Pada penelitian ini, responden yang mengisi kuesioner SUS adalah Pimpinan dan juga Admin KSP. Mengingat aplikasi ini mempunyai dua antarmuka yaitu antarmuka untuk Pimpinan dan juga Admin KSP. Hasil dari kuesioner yang telah diisi oleh responden tertera pada Tabel 6.19 dan terlampir dalam Lampiran I.

Tabel 6. 19 Hasil Pengisian Kuesioner oleh Pimpinan dan Admin KSP

Responden	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
Pimpinan KSP	5	3	5	2	5	1	4	1	4	2
Admin KSP	4	2	5	2	4	2	4	1	4	2

Dari hasil pengisian kuesioner oleh Pimpinan dan Admin KSP, dilakukan perhitungan untuk menentukan total skor responden dimana item bernomor ganjil dinilai dengan cara posisi skala dikurangi satu, sedangkan item bernomor genap dinilai dengan cara lima dikurangi posisi skala. Hasil yang didapatkan dari total pembobotan kemudian dikalikan 2.5. Hasil perhitungan tertera berikut:

Total Skor Responden Pimpinan KSP = $4+2+4+3+4+4+3+4+3+3 = 34$

Total Skor Responden Admin KSP = $3+3+4+3+3+3+3+4+3+3 = 32$

Skor SUS = $(34 + 32) * 2.5 = 165$

Nilai rata-rata = $165/2 = 82,5$

Nilai rata-rata dihitung dengan menggunakan Persamaan 2.8. Dari hasil yang telah didapatkan, dapat disimpulkan bahwa nilai rata-rata yang diberikan oleh kedua responden untuk sistem yang dibuat adalah 82,5, sehingga sistem yang telah dibuat dapat dikategorikan ke dalam sistem yang *acceptable* yang artinya sistem dapat diterima baik oleh KSP Mitra Raya Wates dalam proses pengambilan keputusan pengajuan pinjaman nasabah.

BAB 7 PENUTUP

7.1 Kesimpulan

Setelah dilakukan penelitian berikut dengan pembahasan-pembahasan yang telah dilakukan, dapat disimpulkan sebagai berikut:

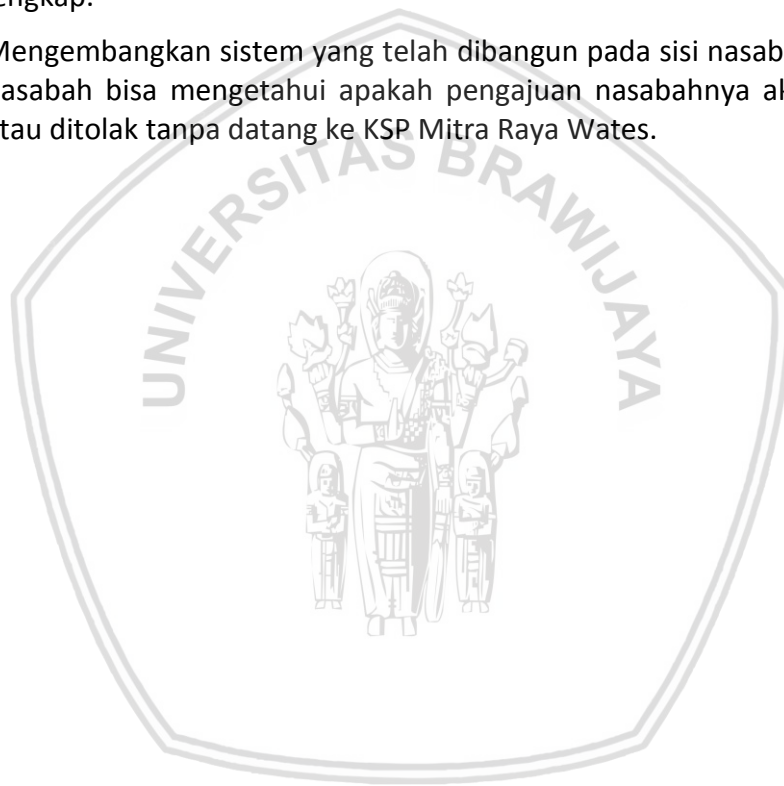
1. Parameter yang menjadi preferensi nasabah untuk memprediksi kredit macet pada KSP Mitra Raya Wates ada tujuh parameter, yaitu jenis kelamin, total pinjaman, jumlah tanggungan, status pernikahan, pendapatan utama, total biaya hidup dan status pinjaman. Tujuh parameter yang digunakan sebagai parameter tersebut berdampak langsung terhadap macet tidaknya pinjaman nasabah, dimana tujuh parameter yang digunakan tersebut terdapat pula pada indikator yang digunakan untuk melakukan survei pada nasabah yang mengajukan pinjaman ke KSP Mitra Raya Wates.
2. Algoritme C4.5 dapat menjadi suatu pilihan dalam menyelesaikan masalah terkait prediksi pengajuan kredit. Metode ini akan menghasilkan *rule* atau aturan berupa pohon keputusan. Untuk menghasilkan sebuah *rule* atau aturan berupa pohon keputusan, diperlukan *pre processing data* terhadap *dataset* yang terkumpul, kemudian dilakukan pembagian data sebesar 80% untuk *data training* dan 20% *data testing*. Setelah dilakukan pembagian data, dilakukan proses perhitungan *entropy* maupun *gain* berdasarkan algoritme C4.5 yang nantinya akan menghasilkan sebuah *rule* atau *aturan* dalam bentuk pohon keputusan yang dapat digunakan untuk memprediksi pengajuan kredit nasabah.
3. Dari hasil pengevaluasian dan pemvalidasian algoritme C4.5 menggunakan *confussion matrix*, didapatkan tingkat akurasi sebesar 94,5946. Sedangkan berdasarkan kurva ROC didapatkan nilai AUC sebesar 0,9689. Sehingga didapatkan kesimpulan jika prediksi kredit macet nasabah pada proses pengajuan pinjaman KSP Mitra Raya Wates dengan menggunakan algoritme C4.5 dapat digolongkan sebagai proses klasifikasi yang sangat baik karena berada pada nilai AUC 0.90 – 1.00.
4. Visualisasi *dashboard* ditampilkan dalam bentuk *diagram pie* yang mendeskripsikan persentase pengajuan ditolak dan diterima, *diagram bar* yang mendeskripsikan *time-series* untuk total pengajuan tiap bulan, dan *diagram line* yang mendeskripsikan *trend* pengajuan yang ditolak maupun yang diterima. Selain itu, terdapat juga *form* yang dapat digunakan baik oleh Pimpinan KSP atau Admin KSP untuk melakukan prediksi pengajuan pinjaman nasabah dan juga pemasukan *dataset* ke sistem. Visualisasi *dashboard* dari sistem juga dilengkapi dengan tabel yang memperlihatkan data pengajuan nasabah beserta dengan preferensinya, tingkat akurasi sistem, dan juga *rule* dalam bentuk *tree* yang dihasilkan dari proses klasifikasi menggunakan algoritme C4.5 dengan *Weka*. Pengujian *usability* yang dilakukan dari visualisasi *dashboard* terhadap responden dari KSP

Mitra Raya Wates menggunakan *System Usability Testing* (SUS) menghasilkan nilai sebesar 82,5. Sehingga dapat disimpulkan jika *dashboard* yang dibangun merupakan *dashboard* dengan kategori *acceptable* dimana sistem dapat diterima baik oleh KSP Mitra Raya Wates dalam proses pengambilan keputusan pengajuan pinjaman nasabah.

7.2 Saran

Saran yang dapat diberikan untuk penelitian selanjutnya adalah sebagai berikut:

1. Menambahkan parameter nasabah yang lain atau parameter eksternal sehingga informasi yang dapat diolah oleh sistem semakin banyak dan lengkap.
2. Mengembangkan sistem yang telah dibangun pada sisi nasabah, sehingga nasabah bisa mengetahui apakah pengajuan nasabahnya akan diterima atau ditolak tanpa datang ke KSP Mitra Raya Wates.



DAFTAR PUSTAKA

- Amin, et. al., 2015. *Implementation of Decision Tree Using C4.5 Algorithm in Decision Making of Loan Application by Debtor (Case Study: Bank Pasar of Yogyakarta Special Region)*. [online] Tersedia di: <<http://ieeexplore.ieee.org/ielx7/7203317/7231384/07231400.pdf>> [Diakses 14 Maret 2018].
- Andriani, A., 2012. *Penerapan Algoritma C4.5 pada Program Klasifikasi Mahasiswa Dropout*. Jakarta: AMIK BSI.
- Anhar. 2010. *PHP & MySql Secara Otodidak*. Jakarta: PT TransMedia.
- Anshari, A., Paryudi, I., & Tjoa A. M., 2013. Performance Comparison between Naïve Bayes, Decision Tree and k-Nearest Neighbor in Searching Alternative Design in an Energy Simulation Tool. *International Journal of Advanced Computer Science and Applications*, vol. 4.
- Baskoro, H., 2010. *Implementasi Algoritma K-Means Menggunakan Data Penyewaan Alat Berat untuk Melakukan Estimasi Nilai Outcome*. S1. Universitas Pembangunan Veteran Jakarta.
- Basuki, A., & Syarif, I., 2004. *Modul Ajar Decision Tree*. Surabaya: PENS-ITS.
- Bauer, D. T., Geurlain, S., & Brown, P. J., 2010. *The Design and Evaluation of Graphical Display for Laboratory Data*. Virginia: University of Virginia.
- Bittner, K. & Spence, I., 2003. *Use Case Modeling*. United States : Pearson Education Inc.
- Booch, et. al., 2005. *The Unified Modeling Language User Guide*. 2nd Edition. United States : Pearson Education Inc.
- Bramer, M., 2007. *Principles of Data Mining*. London: Springer.
- Dendawijaya, L., 2005. *Manajemen Perbankan*. Edisi Kedua, Cetakan Kedua. Bogor: Ghalia Indonesia.
- Dewi, D. P., Djunaidy, P. A., & Pradina, R., 2013. Evaluasi dan Rekomendais Peningkatan Nilai Hidup Pelanggan Berdasarkan Analisis RFM Perilaku Pembelian Pelanggan : Studi Kasus PT. XYZ. *Jurnal Teknik POMITS*, vol. 2.
- Dongming, L., et. al., 2016. *The Application of Decision Tree C4.5 Algorithm to Soil Quality Grade Forecasting Model*. [online] Tersedia di: <<http://ieeexplore.ieee.org/ielx7/7757193/7778858/07778985.pdf>> [Diakses 14 Maret 2018].
- Eckerson, W., 2005. *Deploying Dashboard and Scorecards. TD WI Best Practicies Report*.
- Few, S., 2006. *Information Dashboard Design*. O'Reilly.

- Franico, A., 2018a. *Wawancara Permasalahan Pengajuan Pinjaman pada KSP Mitra Raya Wates Kediri*. Diwawancara oleh Iqbal Taufiq A. N. [wawancara personal] KSP Mitra Raya Wates, 23 Januari 2018, 09.30.
- Franico, A., 2018b. *Wawancara Analisa Kebutuhan pada KSP Mitra Raya Wates Kediri*. Diwawancara oleh Iqbal Taufiq A. N. [wawancara personal] KSP Mitra Raya Wates, 26 Maret 2018, 08.30.
- Gorunescu, F., 2011. *Data Mining: Concepts, Models, and Techniques*. Verlag Berlin Heidelberg: Springer.
- Han, J. & Kamber, M., 2006. *Data Mining Concept and Techniques*. San Fransisco: Morgan Kauffman.
- Hoggson, N. F., 1926. *Banking Through the Ages*. New York: Dodd, Mead & Company.
- Indriati. 2010. *Pengujian Validasi*. [docx] Tersedia di: <<http://indryz.lecture.ub.ac.id/files/2010//Pengujian-Validasi.docx>> [Diakses 6 Juni 2018].
- Jadhav, S. D. & Channe, H. P., 2016. Comparative Study of k-NN, Naïve Bayes, and Decision Tree Classification Techniques. *International Journal of Science and Research*, vol 5.
- Junaedi, H., et. al., 2011. Data Transformation pada Data Mining. *Inovasi dalam Desain dan Teknologi*. pp. 93-99.
- Kamagi, D. H. & Hansun, S., 2014. Implementasi Data Mining dengan Algoritma C4.5 untuk Memprediksi Tingkat Kelulusan Mahasiswa. *Ultimacks*, vol 6.
- Kasmir. 2000, *Management Perbankan*. Jakarta: PT. Rajagrafindo Persada.
- Katore, L. S., Ratnaparkhi, B. S., & Dr. Umale, J. S., 2015. *Novel Professional Career Prediction and Recommendation Method for Individual Through Analytics on Personal Traits Using C4.5 Algorithm*. [online] Tersedia di: <<http://ieeexplore.ieee.org/ielx7/7325162/7342608/07342798.pdf>> [Diakses 14 Maret 2018].
- Kohler, E. L., 1983. *A Dictionary for Accountants*. 6th Edition. New Jersey: Prentice-Hall.
- Kusrini & Luthfi, E. T., 2009. *Algoritma Data Mining*. Yogyakarta: Andi Offset.
- Larose & Daniel, T., 2005. *Discovering Knowledge in Data*. New Jersey: Jhon Willey & Sons, Inc.
- Liao. 2007. *Recent Advances in Data Mining of Enterprise Data Algoritms and Application*. Singapura: World Scientific Publishing.
- Malik, S., 2005. *Enterprise Dashboard – Design and Best Practices for IT*. New Jersey: Jhon Willey & Sons, Inc.

- Maurina, D., 2015. *Penerapan Data Mining untuk Rekomendasi Beasiswa pada SMA Muhammadiyah Gubug Menggunakan Algoritma C4.5*. S1. Universitas Dian Nuswantoro.
- Meissler, D., 2010. *A List of Different Case Types*. [online] Tersedia di: <<https://danielmiessler.com/blog/a-list-of-different-case-types/>> [Diakses 4 Juni 2018].
- Menarianti, I., 2015. Klasifikasi Data Mining dalam Menentukan Pemberian Kredit Bagi Nasabah Koperasi. *Jurnal Ilmiah Teknosains*, Vol. 1, No. 1.
- Nielsen, J., 2012. *Usability 101: Introduction to Usability*. [online] Tersedia di: <<https://www.nngroup.com/articles/usability-101-introduction-to-usability/>> [Diakses 22 Juni 2018].
- Oktaviana, A. R., 2016. *Penerapan Data Mining Klasifikasi Pola Nasabah Menggunakan Algoritma C4.5 pada Bank BRI Batang*. Skripsi. Universitas Dian Nuswantoro.
- Orts, D., 2005. *Dashboard Development and Deployment, New Business*. [online] Tersedia di: <<http://news-business.vlex.com/vid/dashboarddevelopment-and-deployment-62243694>> [Diakses 26 Januari 2016].
- Prasetyo, E., 2014. *Data Mining Mengolah Data Menjadi Informasi Menggunakan Matlab*. Yogyakarta: Andi Yogyakarta.
- Pramudiono, I., 2003. *Pengantar Data Mining: Menambang Permata Pengetahuan di Gunung Data*. Ilmu Komputer.com.
- Pressman, R. S., 2010. *Software Engineering A Practitioner's Approach*. 7th Edition. New York : McGraw-Hill.
- Riyanto, S., 2009. *Membuat Web Portal Multi Bahasa Joomla*. Jakarta: PT Elex Media Komputindo.
- Rosmilia, R., 2009. *Pelaksanaan Penyelesaian Kredit Bermasalah*. Thesis. Universitas Diponegoro.
- Santosa, B., 2007. *Data Mining Teknik Pengumpulan Data Untuk Keperluan Bisnis*. Yogyakarta: Graha Ilmu.
- Sauro, Jeff. 2011. *Measuring Usability With The System Usability Scale (SUS)*. [online] Tersedia di: <<http://www.measuringu.com/sus.php>> [Diakses 22 Juni 2018].
- Sauro, Jeff. 2013. *A Single-Item Measure of Website Usability: Comments on Christophersen and Konradt (2011) Interacting With Computers, Special Issue: Commentary on Scale Derivation*. 25 (4): 302 – 303.
- Shalahuddin, M. & S, R. A., 2015. *Rekayasa Perangkat Lunak : Terstruktur dan Berorientasi Obyek*. s.l.:Informatika.
- Sibero, A. F. K., 2011. *Kitab Suci Web Programming*. Yogyakarta: Mediakom.
- Soleh, A. Z., 2005. *Ilmu Statistika*, Cetakan Pertama. Bandung: Rekayasa Sains.

- Sucipto, A., 2015. Prediksi Kredit Macet Melalui Perilaku Nasabah pada Koperasi Simpan Pinjam dengan Menggunakan Metode Algoritma Klasifikasi C4.5. *Jurnal DISPROTEK*, vol. 6.
- Sunjana. 2010. Aplikasi Mining Data Mahasiswa dengan Metode Klasifikasi Decision Tree. *Seminar Nasional Aplikasi Teknologi Informasi 2010*, 24-29.
- Supono & Putratama, V., 2016. *Pemrograman Web dengan Menggunakan PHP dan Framework Codeigneter*. Yogyakarta: Deepublish.
- Susanto, E., 2012. *Data Mining Menggunakan Weka*. [online] Tersedia di: <<http://www.erdisusanto.com/2012/06/data-mining-menggunakan-weka.html>> [Diakses 4 Juni 2018].
- Tan, S., Kumar, P., & Steinbach, M., 2005. *Introduction to Data Mining*. London: Addison-Weasley.
- Utdirartatmo, F., 2005. *Teori Bahasa dan Otomata*. Yogyakarta: Graha Ilmu.
- Witten, Ian H., et. al., 2016. *The Weka Workbench, Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques*. 4th Edition. San Fransisco: Morgan Kauffman.

